DESIGN AND IMPLEMENTATION OF AN AUTOMATED EXAMINATION SYSTEM

A MASTER'S THESIS

in

Computer Engineering
Middle East Technical University

By
Mustafa TÜTÜNCÜ
September 1987

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Halim Doğrusöz
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. A.Ziya Aktaş
Chairman of the Department

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Bülent Epir
Supervisor

Examining Committee in Charge

Assoc. Prof. Dr. Müren Gökeri (Chairman)
Assoc. Prof. Dr. Ünal Yarımağan
Asst. Prof. Dr. Faruk Tokdemir
Asst. Prof. Dr. İbrahim Akman
Asst. Prof. Bülent Epir

ABSTRACT

DESIGN AND IMPLEMENTATION OF AN
AUTOMATED EXAMINATION SYSTEM

TÜTÜNCÜ, Mustafa
M.S. in Computer Eng.
Supervisor: Asst.Prof. Bülent Epir
September 1987, 193 pages

In this study, an Examination System which is one of the fundamental parts of an Examination Organization System to give an examination for a population, is designed and implemented. The main functions of the Examination System are determining the places where the candidates take the examination, evaluation of the examination and handling all printouts related to the examination.

Developing this study, a structured approach is used to design of the Examination System. For this purpose the logical design of the system is actualized by examining the existing ÖSYM applications in detail. Then, the physical design of the Examination System is carried out and the system is implemented.

During the implementation of the system, a software is developed as a multi-user which can operate in interactive environment. This software can communicate with the Registration System which is the other fundamental part of the Examination Organization System and both of them form an information system titled REX.

Key Words: Structured System Analysis, Examination Organization, Examination System, Evaluation of the Examination

ÖZET

ÖZDEVİNİMLİ BİR SINAV SİSTEMİNİN TASARIMI VE GERÇEKLEŞTİRİMİ

TÜTÜNCÜ, Mustafa
Yüksek Lisans Tezi, Bilgisayar Müh. Bölümü
Tez Yöneticisi: Y.Doç. Bülent Epir
Eylül 1987, 193 sahife

Bu çalışmada, bir kitleye sınav vermek amacıyla uygulanacak Sınav Organizasyonunun temel bölümlerinden birisi olan Sınav Sistemi tasarlanmış ve gerçekleştirilmiştir. Sınav Sisteminin temel işlevleri, sınava katılan adayların nerede sinava gireceklerini belirlemek, sınavı değerlendirmek ve sınavla ilgili tüm döküm işlerini ele almaktır.

Sınav Sistemini tasarlamak için yapılan çalışmada yapısal sistem tasarımı yaklaşımı kullanılmıştır. Bu amaçla ÖSYM'deki mevcut uygulama ayrıntılarıyla incelenerek gerçekleştirilecek sistemin mantıksal tasarımı yapılmıştır. Daha sonra Sınav Sisteminin fiziksel tasarımı ele alınmış ve sistem gerçekleştirilmiştir.

Sistem gerçekleştirilirken, etkileşimli ortamda çok kullanıcıyla çalışabilen bir yazılım geliştirilmiştir. Bu yazılım, sınav organizasyonunun diğer temel bölümünü oluşturan Kayıt Sistemi ile iletişim kurabilmekte ve birlikte bir bilişim sistemi oluşturmaktadır. Bu sisteme "Registration and EXamination system" adı verilmiş ve REX olarak kısaltılmıştır.

Anahtar sözcükler: Yapısal Sistem Çözümleme, Sınav Organizasyonu, Sınav Sistemi Sınav Değerlendirme.

ACKNOWLEDGEMENT

The writer expresses his sincere gratitude to Asst.Prof. Bülent Epir for his excellent guidance and patience in directing this study. Deep appreciation is extended to Assoc.Prof.Dr. Ünal Yarımağan for his valuable suggestions and counselling. Special appreciation is also extended to Prof.Dr. Altan Günalp for his permission to implement this study in ÖSYM and to use the resources of the center. Thanks also due to my wife, Gönül, for her unfailing support and encouragement. The writer also wishes to express his gratefulness to Mehmet Altunay for his assistance. The following people have had a direct interest in this study which was helpful: Dr. Atalay Yunusoğlu, Giray Berberoğlu, Suat Canpolat, Eyüp Onat and Halim Tatar. Finally, I would like to thank my colleague Argun Atasagun for his cooperation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Registration and Examination concepts are terms being frequently encountered in daily life. In order to develop an information system based on these concepts, two master theses studies complementing each other have been implemented. These theses are designed for the applications in the Student Selection and Placement Center (ÖSYM) in Turkey.

The main purpose of ÖSYM is selection and placement of students for higher education institutions in Turkey. For this purpose a project named "Student Selection and Placement System (ÖSYS)" is being carried out [ÖSYM, 198Ø] [ÖSYM, 1984] [PAYASLIOĞLU, A., 1985] [TOKER, F., GÜNALP A., 1982]. In addition to this basic project, ÖSYM carries out many other projects. Most of these projects are the requests of examination organizations coming from various institutions.

These projects are being carried out by adapting the programs which were developed for the ÖSYS project. This project requires a complex organization in addition to systematic and fairly ordered data processing methods and programs. For this reason, it is not suitable to use a project directed to a specific application, when the examination organizations which have different properties.

Besides that, the existing examination organization is executed by various subunits whose duties are predefined [ÖSYM, 1985a]. However, in this structure, any failure in one of these subunits, effects the entire project. That is to say, the subunits are too much dependent on each other. Whereas, in this kind of project, the subunits must be independent as much as possible and must be conceptually whole. That means, the subunits must be functionally cohesive [AKTAŞ, A.Z., 1987] [BROOKS, C.H.P. et al., 1982] [WEINBERG, V., 198Ø].

For this purpose, it was necessary to implement an information system which is modular, general purpose, structured and functionally cohesive that can be used in any type of examination organizations within the ÖSYM.

Taking these requirements as a starting point, phases of the information system life cycle which are analysis, logical design, physical design and implementation have been followed [AKTAŞ, A.Z., 1987] [BROOKS, C.H.P. et al., 1982] [BURCH, J.G., STRATER, F.R., 1984] [WEINBERG, V., 198Ø]. During understanding the existing system, many documents prepared by ÖSYM for various examination organizations have been analyzed [ÖSYM, 1985a] [ÖSYM, 1985b] [ÖSYM, 1985c] and the proposed system has been developed by means of two master theses.

The first master thesis, which discusses the registration of the candidates and determination of the executive staff necessary for the examination is titled "Design and Implementation of an Automated Registration System" [ATASAGUN, C.A., 1987].

The second master thesis, which complements the first one and discusses on the prearrangements and evaluation of the examination is titled "Design and Implementation of an Automated Examination System".

These two theses together are forming an information system titled "Registration and Examination System" (REX).

1.1. Purpose and Scope of REX

The purpose of REX is to develop software that implements the system which will perform the examination organization automatically beginning from the application of the candidates up to the determination and announcement of the examination results.

REX improves the ÖSYS project as it was developed as a package program which can be used for every kind of examination organization of ÖSYM.

During the development of the REX software a structured system analysis study was performed. Data Flow Diagram (DFD) and Warnier-Orr Diagrams were selected as tools [AKTAŞ, A.Z., 1987] [BROOKS, C.H.P. et al., 1982] [WEINBERG, V., 198Ø].

As a result of the analysis of the existing system, some new concepts were developed and the proposed Examination Organization System was constructed on these concepts. These concepts are the user name which identifies the examination, the examination centers in which the examination will be performed, the sessions which are the parts of the examination, the areas which are the study groups and the subareas which are the group of questions in the area.

The subsystems which comprise the Examination Organization System are: Registration Supporting Subsystem, Examination Supporting Subsystem, Test Preparation Subsystem and REX Subsystem. Supporting Subsystems are the interfaces between external environment and REX Subsystem. Test Preparation Subsystem produces the tests which will be used in the examination. REX Subsystem is the software section which processes the the data and produces the necessary printouts of an examination organization.

REX consists of subsystems that may transfer data one to another. Some of the programs in the subsystems are of specific, and others of general purpose type. The general purpose programs are not interpreters. They are software that have the properties of "program writing programs", which run in an interactive environment. All the programs guide the users with menus at each stage while the system is running.

1.2. Subsystems of REX

REX consists of three main subsystems.

i)Registration Subsystem: A Candidate File is created from the information given by the candidates having applied for admission to the examination. Some information controls are performed on this file in order to correct the errors. Staff Files are created to record the information belonging to the persons that will be charged with the responsibility of the examination. Using the information of these files the necessary actions are performed.

ii)Examination Subsystem: Using the information of the Candidate File, the Building, Room and Examination Files are created. The rooms in which the candidates will take the examination are determined. Necessary actions are performed to transfer the documents belonging to the staff which was determined by the Registration Subsystem and to transfer the documents to be used in the examination. After the examination, the evaluation process is performed and the results are obtained.

iii)Supporting Subsystem: The purpose of this subsystem is to perform the necessary activities to ensure the use of system program packages. It also takes the necessary security measures and impedes the formation of inconsistent structures.

1.3. Implementation Notes on REX

REX is developed on Burroughs mainframe systems and since many parts of the software are machine dependent it is not portable.

Generally REX is implemented by the PL/I programming language. In some cases ALGOL is used to support REX, especially to syncronize the programs which can be run as independent tasks.

Classic file structures and access methods have been used.

According to the characteristics of the application some special algorithms have been developed.

The total size of the software is more than 5000 statements.

Approximately two years have been spent developing this system.

2. EXAMINATION AND REGISTRATION CONCEPTS

Since REX is closely related to the registration and examination concepts, general definitions of these concepts will be presented.

2.1. Examination Concept

The process of determining the aptitude of individuals within a community, relative to each other, in one or more subject areas, or determining the aptitude of each individual within the community, independently of the other ones, according to some predefined level, is called examination.

Examinations are performed for two purposes.

i) To select

ii) To measure

In the examinations in which the purpose is selection, the answers given by each of the candidates are graded and the scores put in a descending order. According to the purpose of the examination, a specified number of candidates are designated successful beginning from the highest score. The score of the last candidate in the list is called "lower limit score". The lower limit score has the following properties.

1)This score can be determined only after the scores of all the candidates have been calculated.

2)When an examination consisting of the same questions is applied to two different communities, the obtained scores will be different because of dissimilarities between the levels of knowledge of the people entering the examination.

In the examinations in which the purpose is to measure, the lower limit score is predetermined. This score is compared with the score obtained from the answers of each candidate and the ones who can not reach the limit are deemed to be unsuccessful.

The point which must be emphasised here is that the lower limit score is determined after the examination in selection type examinations and before the examination in measurement type examinations.

Basically, examinations are performed by using one or more of the following methods.

i) Written
ii) Oral
iii) Applied

Written examinations may be either essay or fixed-answer type. REX is designed for fixed-answer type examinations [PAYASLIOĞLU, A., 1985] [ÖZÇELİK, D.A., 1981]. The reasons are as follows.

-This is the most practical method to be applied for examinations having a large number of applicants like in the case of ÖSYS application in Turkey.

-Test type examinations are more suitable for measuring the knowledge and skills of candidates as they are more objective, standard and reliable when compared with essay type examinations.

-Test results can be evaluated rapidly and correctly via optical readers and computers.

2.2. The Registration Concept

In order to carry out the organization of an examination, it is necessary to determine the candidates who will take the examination. Application forms which are designed for this purpose are sent to the candidates. The candidates make their application by filling these forms.

The process of recording the information about candidates obtained from the application forms to be used for examination or other purposes is called, the registration.

The registration process can be handled in two ways depending on whether the candidates to be registered is known or not.

i)If the candidates to be registered is known then the application forms are prepared only for these candidates and sent to them. Renewal registrations in the universities can be an example for this type of registrations.

ii)If the candidates to be registered is not known then the candidates take the application forms from predefined centers and the ones who fill and give the forms are said to be registered. University selection examination registrations can be an example for this type of registrations.

The registration process can be handled in two ways if the application periods are considered.

i)The registrations in which the application period is predetermined. The candidates must apply within the predetermined application period. Registrations for employment exams of firms can be an example for this type of registrations.

ii)The registrations in which the application period is not defined but comes into force after the application of the candidate. Registrations for drivers licence applications can be an example for this type of registrations.

3. PROPOSED EXAMINATION ORGANIZATION SYSTEM

Proposed Examination Organization System schema is shown in Figure 3.1 by using Data Flow Diagrams (DFD) together with the files, sources/sinks and data flows. All files, sources/sinks and data flows are described in detail in the following sections.

3.1. Files

All the files used in REX are described below.

Candidate File:It is the file in which the information, concerning the candidates having applied to take the examination, are stored. This file is used, to record the information gathered with the Application Form from the candidates by the Registration Subsystem; to transfer the assignment information gathered by the Examination Subsystem prior the examination and the evaluations information after the examination.

Examination file:This file is created by selecting the candidates that will be admitted to the examination from the Candidate File, after the completion of the registration. The evaluation of the examination and the determination of the scores is performed by using this file. The scores obtained after the evaluation, are transferred to the Candidate File.

Room Files:These are the files that has been formed by choosing sufficient number of rooms, from the General Room File, for the candidates that will be admitted to the examination. There is one Room File for each session of the examination. Room Files are created and used during the room assignment.

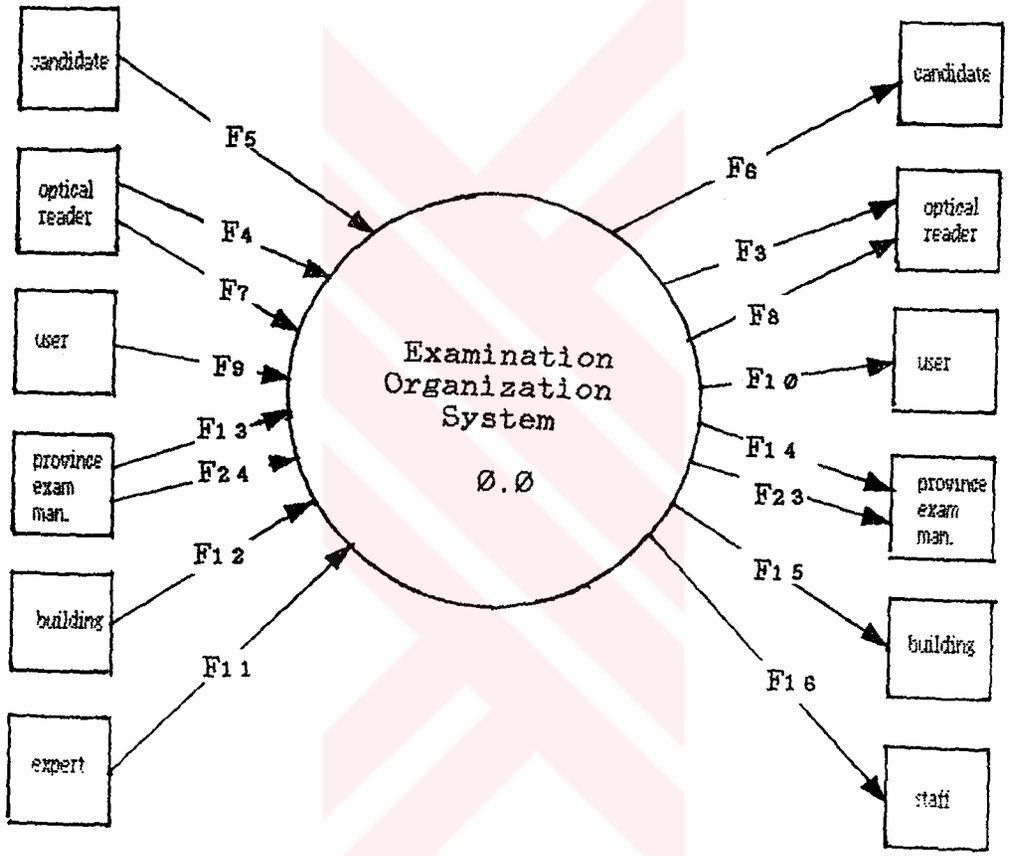


Figure 3.1 DFD of Examination Organization System

Staff Files:The staff files consist of the information concerning the Proctor, Reserve Proctor, Room Director, Building Examination Director, Assistant Building Examination Director and Building Manager appointed in the examination. The staff appearing in these files are determined by the Province Examination Manager. There is one Staff File for each session of the examination. Staff Files are created and used during the staff assignment.

Building File:It is the file in which all the information about the buildings utilized in the examination are stored.

Optical Files:These are the magnetic tape files formed in order to transfer the information of the optical forms read on an Optical Reader to REX for registration, examination or any other purpose.

3.2. Sources and Sinks

Candidate:Is the person who wishes to take the examination. The candidates make applications for registration. The places in which they will take the examination and results are declared to the candidates.

User:The person who uses REX is called the user. The user enters commands to REX. In return, REX provides reports on performed studies and guides the user with menus.

Province Examination Management:It is the center which lends the examination documents to the buildings the documents related with the staff to them; and which sends back the examination documents from the buildings.

Building:They are the centers where the documents coming from the Province Examination Management are distributed and collected.

Staff:They are the people who are responsible from the examination execution. They get Duty Assignment Form, Identification Card and Staff Cheque from the Province Examination Management.

Expert:They are the persons who have specialized on their areas and who prepare the questions used in the examination. The examinations which are prepared by these persons, are sent to the Province Examination Management.

Optical Reader:They are the electronic devices on which the optical readings of Application Forms and Answer Sheets are performed.

3.3. Data Flows

F_i representation has been used on the DFD while data flows were being coded. The meanings of data flows are as follows.

F_1 :The data transferred to the Optical File from Optical Reader.

F_2 :The data transferred from the Optical File to the Examination Organization System.

F_3 :Application Forms which are to be read by means of an Optical Reader.

F_4 :The Application Forms which have been damaged during the optical reading process.

F_5 :The Application Forms received from the candidates.

F_6 :The Application Forms, Examination Entrance Forms and Examination Results Forms which are sent to the candidates.

F7 : Faulty Answer Sheets.

F8 : Answer Sheets which are to be read by means of an Optical Reader.

F9 : The commands that the user enters to the system to inform about his/her demand.

F10 : The responses, reports and menus that the system offers to the user.

F11 : The questions prepared by the experts to be used in the examination.

F12 : The examination documents which are to be sent to the system from the building after the examination.

F13 : The examination documents which are to be sent to the system from the Province Examination Management after the examination.

F14 : The examination documents to be used in examination process.

F15 : The examination documents which are sent from the system to the buildings.

F16 : The documents which are sent from the system to the examination staff.

F17 : Information about the examination staff.

F18 : Information about the buildings that are used for the examination.

F19 : Information about the rooms that are used for the examination.

F_{20}: Information about the candidates related to the evaluation of the examination.

F_{21}: Information about the registered candidates.

F_{22}: Information which defines and controls the examination organization.

F_{23}: Staff Notification Schedules sent to the Province Examination Management.

F_{24}: Staff Notification Schedules returned from the Province Examination Management.

3.4. The Subsystems of the Examination Organization System

The subsystems which form an Examination Organization System, files and sources/sinks related with these subsystems and data flows between them are shown in Figure 3.2 by using DFD.

3.4.1. Registration Supporting Subsystem

It is designed to form a Registration Supporting Subsystem which is formed by manpower and assisting machines to examine the documents which come from the candidates, to correct and determine the faulty cases, to transfer the information on these documents to REX, to regulate and send the documents prepared by REX for candidates. It also sends the Staff Notification Schedules to the Province Examination Management and transfers the information about the staff to the REX. This Supporting Subsystem is communicating with Optical Reader, Province Examination Management and Optical File.

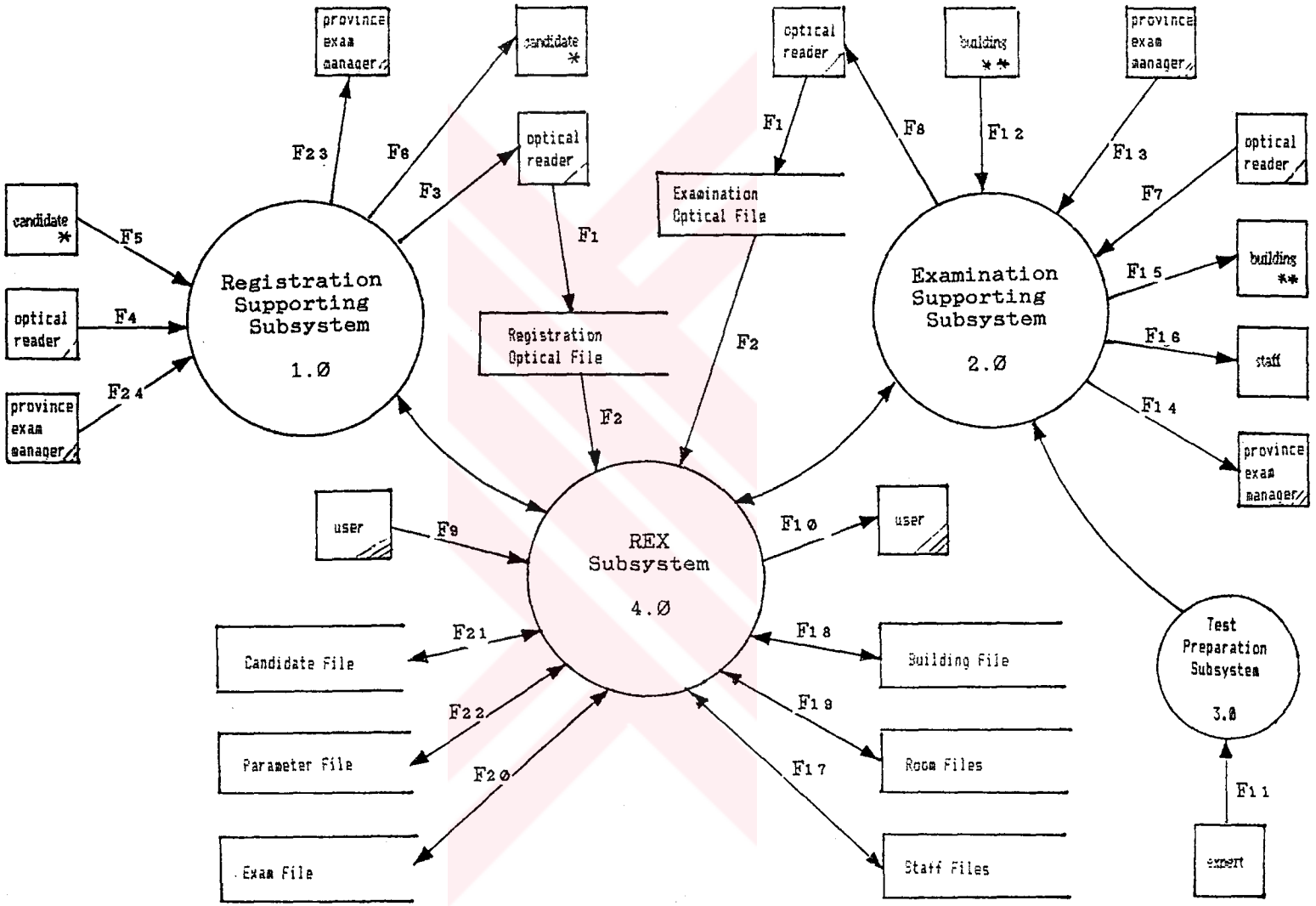


Figure 3.2 DFD of Subsystems of Examination Organization

3.4.2. Examination Supporting Subsystem

It is designed to form an Examination Supporting Subsystem which is formed by manpower assisting and machines to prepare all printed documents to be sent to the Province Examination Management before the examination, to receive the documents after examination and to transfer the candidates answers to REX. This Supporting Subsystem is communicating with the Province Examination Management, Building, Staff, Optical Reader and Optical File.

3.4.3. Test Preparing Subsystem

It is the subsystem which prepares and prints the questions to be used in the examination, in the booklet form. This subsystem transfers the question booklets to the Examination Supporting Subsystem to be sent to the Province Examination Management.

3.4.4. REX Subsystem

This is the subsystem in which all data processing activities related to the examination organization is carried out. This subsystem processes the information coming from the supporting subsystems to obtain necessary printouts. These printouts are transferred to sources/sinks via the Supporting Subsystems.

The subsystems forming REX, files and sources/sinks related with these subsystems and data flows between them are shown in Figure 3.3 by DFD.

In this section, the concepts, subsystems, implementation, properties and requirements of REX will be explained.

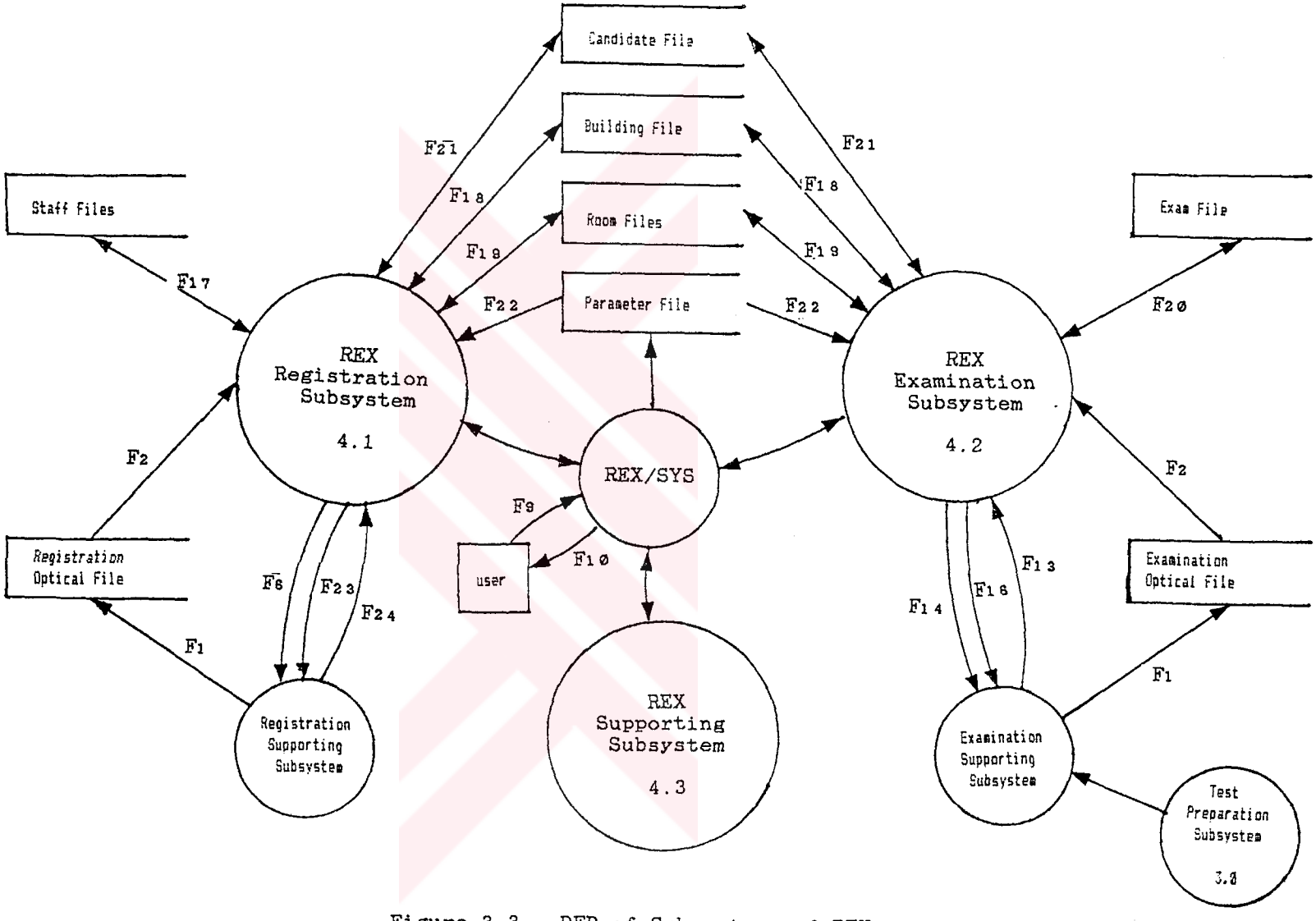


Figure 3.3 DFD of Subsystems of REX

3.4.4.1. Concepts of REX

As a result of the analysis of the existing system it has been conceived that the implementation of a general purpose examination organization may be settled on the following five concepts.

i)User system name which identifies the examination

ii)Number of centers in which the examination will be performed

iii)Number of sessions in the examination

iv)Number of areas in each session

v)Number of subareas in each area

The information system, which takes these aspects on a general purpose basis, will be of structured and modular type. Since these concepts are forming a basis for the information system, they have been analyzed in details.

3.4.4.1.1. User System

All data files and user programs which are formed by REX's studies on an examination organization are kept as a directory, under a certain name. Its reason is to distinguish the structures of each examination while working on more than one examination organizations at the same time. The name that defines on examination organization is called the User System. This name consist of five alphanumeric characters.

These names may be meaningless, such as ABX2Y. However, it shall be more useful to use abbreviated names indicating the performed examination organization and generally known for this examination.

ÖSS for Student Selection Examination, ÖYS for Student Placement Examination, AÖF for student examinations of Open Education Faculty, YÖS for Foreign Student Examination, SSYB for Ministry of Health and Social Assistance examination can be examples for User System names.

3.4.4.1.2. Examination Center

This is the residence center as province, district etc. in which the examination will be performed. The examination organization is executed in the same manner in each center. A Province Examination Manager is present in each center and is responsible to execute the examination in conformity with the rules and to take the precautions that will resolve the problems that may arise.

3.4.4.1.3. Session

These are the parts of an examination which are performed in different time portions. According to its purpose, an examination may be performed either in one session or in more than one session. In some examinations, the previously performed sessions are used to determine the candidates that will be admitted to the following ones. In those type of examinations the candidates may be allowed to enter in more than one session according to their success. In some other type of examinations the candidates may be admitted into more than one session, without any condition and according to the areas choosen by the candidate.

3.4.4.1.4. Area

The study field, for which it is necessary to prepare different questions is named area. Some of these areas are forming a group according to the subjects they are consisting. The property of these groups is that, the candidates have to take the examination only in one of the subject within the area. Though the candidates may choose one of the group in the area, they may take the examination in more than one area by choosing among the groups of some other areas. For this reason it is not possible to perform an examination for two different groups at the same time. The part of the examination reserved for one group is executed in one session.

3.4.4.1.5. Subarea

The group of questions which are of specific type within the questions of a area is called subarea. The candidate who takes an examination in one area is considered to be admitted to all of the subareas within this area.

3.4.4.2. Relations Between The Concepts

-The user system name identifies the examination.

-The examination is performed in one or more than one center.

-The examination is executed in one or more than one session in each center.

-In each session there is one or more than one area which the candidate may take the examination in only one of them.

-Each area may be divided into subareas.

-The number of subareas are equal for each of the areas in one session.

The relations between the concepts are shown in Figure 3.4.

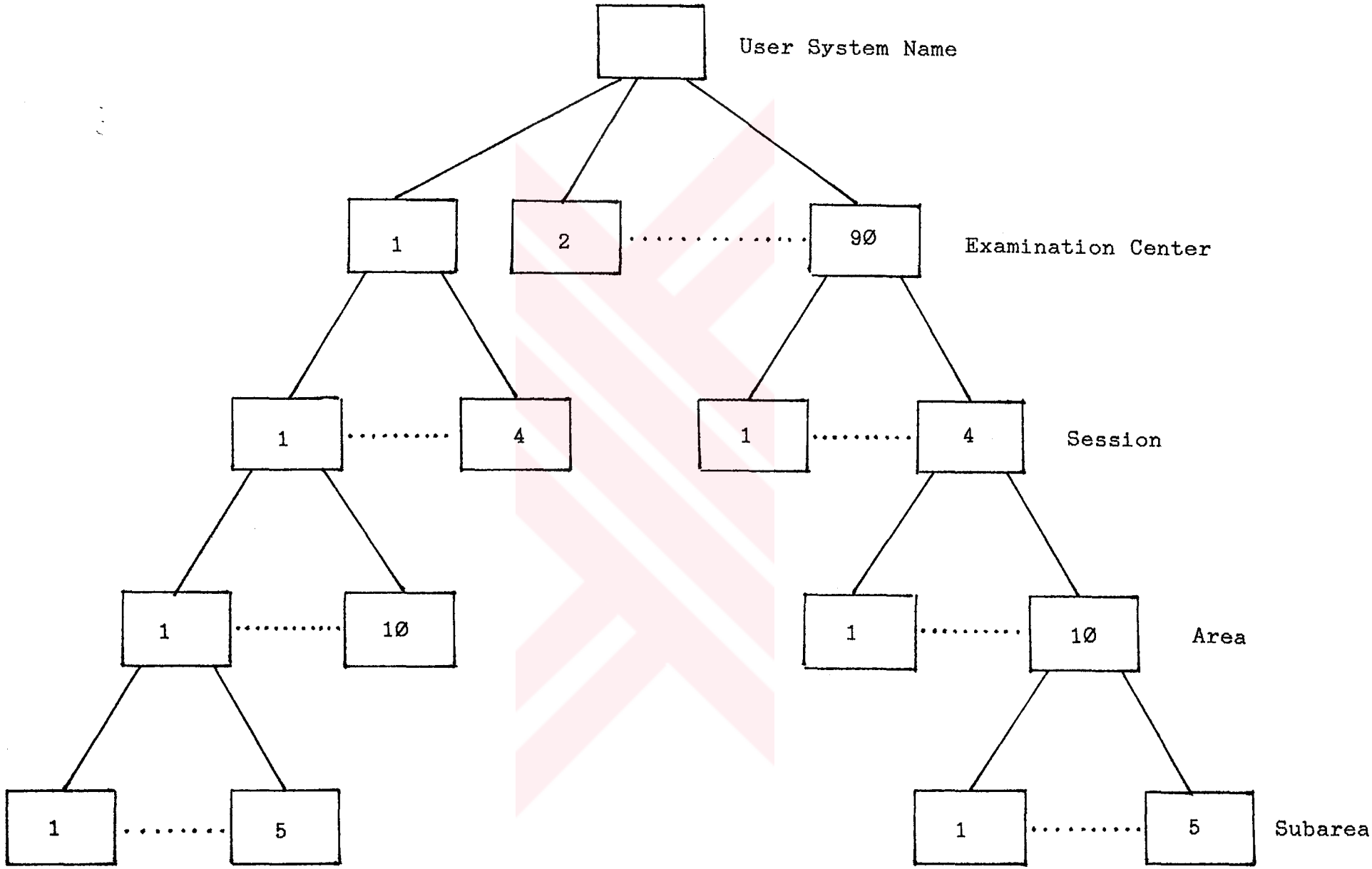


Figure 3.4 Relations between the REX Concepts

It will be useful to examine the four concepts that their descriptions and relations to each other is mentioned above, on the actual applications at ÖSYM.

i) Application 1

In this application, the examination which is utilized for foreign students in the admittance of higher education programs (YÖS) shall be introduced.

This examination is performed in seven centers, six of them are in Turkey.

It is performed in one session.

All the candidates take the same examination, it is to say, there is one area.

This area is divided into two subareas.
- Basic Learning Skills test
- Turkish Language Proficiency test

ii) Application 2

Assistanship examination of Ministry of Health and Social Assistance shall be introduced in this application.

This examination is performed in Ankara, only in one center.

It is applied in two sessions, one is on Saturday and the other on Sunday.

In the first session, there are six areas as English, French, German, Italian, Spanish and Russian.

In the second session, there are three areas as General Medicine, General Microbiology and General Biochemistry.

All the candidates take one of the areas of the first session. They can be admitted for the second session only if they have succeeded in the first session's test.

In this application there are no subareas.

iii) Application 3

In this application, the student placement examination (ÖYS) which is performed for placing students to Turkish higher education programs, shall be introduced.

This examination is performed presently in twenty one examination centers, twenty of them are in Turkish provinces, one is in Lefkoşa, KKTC.

It is applied in one session.

All the candidates take the same examination, that is to say, there is only one area. This area is divided into five subareas as Turkish, Mathematics, Science, Social Science and Foreign Language.

iv) Application 4

In this application, the midterm, final and make-up examinations of Eskişehir Anadolu University Open Education Faculty's (AÖF) students, shall be introduced.

Instruction period of AÖF is four years. The faculty has two programs as Economics and Business Administration. It is required for the students to take eight courses in each year. In the first two years, all

the students have to take the general courses without distinction. Program distinctions take place in third and fourth years. In these classes, five courses are taken in common and three courses with program separations. So, there is a total of thirty eight courses to be studied.

Promotion rule is applied in AÖF. To be able to promote, a student has to succeed all courses or must have been failed in maximum two of the courses. A student who has failed in one or two courses is sentenced to be on probation and takes these courses together with the eight courses of the next class. A student who has failed in more than two courses, repeats these courses. In this case, minimum three, maximum ten courses can be taken. This rule excludes only last year students. A last year student who has failed in one or two courses can not promote to a higher class, so, he/she takes one or two courses in the next year. A student who has completed all courses, becomes a graduate. If no amnesty law is applied, a student who has failed twice in one course without any intervals is dismissed.

After these explanations, the concepts given for REX can be considered as follows for this application.

There are eighteen centers for this examination.

The examination is applied in four sessions as on Saturday morning, Saturday afternoon, Sunday morning and Sunday afternoon.

First and third year students take examinations for the first four courses on Saturday morning and the rest four courses on Sunday morning, second and fourth year students take examinations for the first four courses on Saturday afternoon and the rest four courses on Sunday afternoon.

All students take the examination in two sessions. However, the students who are on probation from the previous class, can be exposed to participate the third and fourth sessions for their probational courses. For the same reason, repeat students, can take the examination in one session, if the related courses are given in the same session. In summary, the students can take the examination in maximum four sessions, depending on their courses.

A total of twelve areas exist in these four sessions. These areas and sessions are as follows.

1)First year, first four courses: Saturday morning.

2)First year, second four courses: Sunday morning.

3)Second year, first four courses: Saturday afternoon.

4)Second year, second four courses: Sunday afternoon.

5)Third year, first four courses in economics: Saturday morning.

6)Third year, second four courses in economics: Sunday morning.

7)Third year, first four courses in business administration: Saturday morning.

8)Third year, second four courses in business administration: Sunday morning.

9)Fourth year, first four courses in economics: Saturday afternoon.

1Ø)Fourth year, second four courses in economics: Sunday afternoon.

11)Fourth year, first four courses in business administration: Saturday afternoon.

12)Fourth year, second four courses in business administration: Sunday afternoon.

3.4.4.3. Subsystems of REX

3.4.4.3.1 Registration Subsystem

The actions executed by this subsystem are listed below.

-It forms the candidate file depending on the information gathered from the candidates and to the parameters showing the number of center, area and session in which the examination will be executed.

-It prints the Application Forms.

-It records the candidate information to the file.

-It prints the control listings.

-It records the updates caused by faulty information.

-It supplies the printouts and the statistical schedules for the future stages at the completion of the transactions on the candidate file.

-It warns the Examination Supporting Subsystem for the printing of the Examination Entrance Form and Examination Results Form.

-It sends the list of all the buildings and rooms to be used in the examination in order to determine the staff to the Province Examination Management.

-It forms the staff file using the information concerning the staff.

-It prints all the documents related to the candidates and the staff.

3.4.4.3.2. Examination Subsystem

The actions executed by this subsystem are listed below.

-It forms the Building, Room and Examination Files.

-It makes the room assignment using the distributions taken from the candidate file delivered by the Registration Subsystem.

-It prints all the documents related to execution of the examination.

-It transfers the information concerning the places in which the candidates have taken the examination to the candidate file, at the end of the room assignment.

-It supplies the raw, standard and weighted scores by evaluating the examination.

-It transfers the examination results to the candidate file.

3.4.4.3.3. Supporting Subsystem

The actions executed by this subsystem are listed below.

-It supplies the necessary logs.

-It takes the necessary precautions towards the facts that may arise in case of a rupture.

-It warns the user when a rupture has occured during the previous operations and lead the system to the first encountered unfaulty point.

-It takes the necessary measures in order to ensure the security in the file accessing and program operating levels, while the system is used.

-It ensures the operation of some program packages concerning REX in the system.

-It performs the works like copying, deleting, listing, checking the presence, analyzing in an interactive environment, changing the name of the files organized with REX.

3.4.4.4. Physical Implementation of REX

Rex consists of three modules which run under the control of a main program. The name of the main program in the system is REX/SYS. The three modules start the Registration, Examination, Supporting systems and they are named as REX/REGISTRATION, REX/EXAMINATION, REX/SUPPORTING respectively.

When REX is used, the main program is activated and according to the choice of the user, one of the three modules is run. When one of the modules began to run, the main program enters to the waiting state. When the module has terminated its task, the main program continues to run. The module which is run under the control of the main program may activate another submodule if necessary. In this case the module activated from the main program will wait till the task of the submodule is terminated. So the main program waits the module, and the module waits the submodule. The schema concerning this case is shown in Figure 3.5.

3.4.4.5. How to Run REX

The user who wants to activate REX, must initiate a session with the usercode REX and run the program REX/SYS from a terminal.

When REX/SYS is run, the form/menu shown in Figure 3.6 is displayed on the screen. On the fields of this form/menu, the <user system name> and the <password> must be entered. However, the user may want to see the present user systems by pressing the SPCFY key. In this case, using the form given in Figure 3.7 all the user system names are displayed on the screen. It is also possible to terminate the session by entering the word QUIT.

When a <user system name> and a <password> is entered, it is checked whether this user system has been previously created or not. If it has been previously created, then the session is directly started under this user system. In this stage, the <station number> is added to the Parameter File named <user system name>/PARAMS and this file is copied with the name <station number>/PARAMS. This is done, in order to get the general information about the user system, by means of the <station number>.

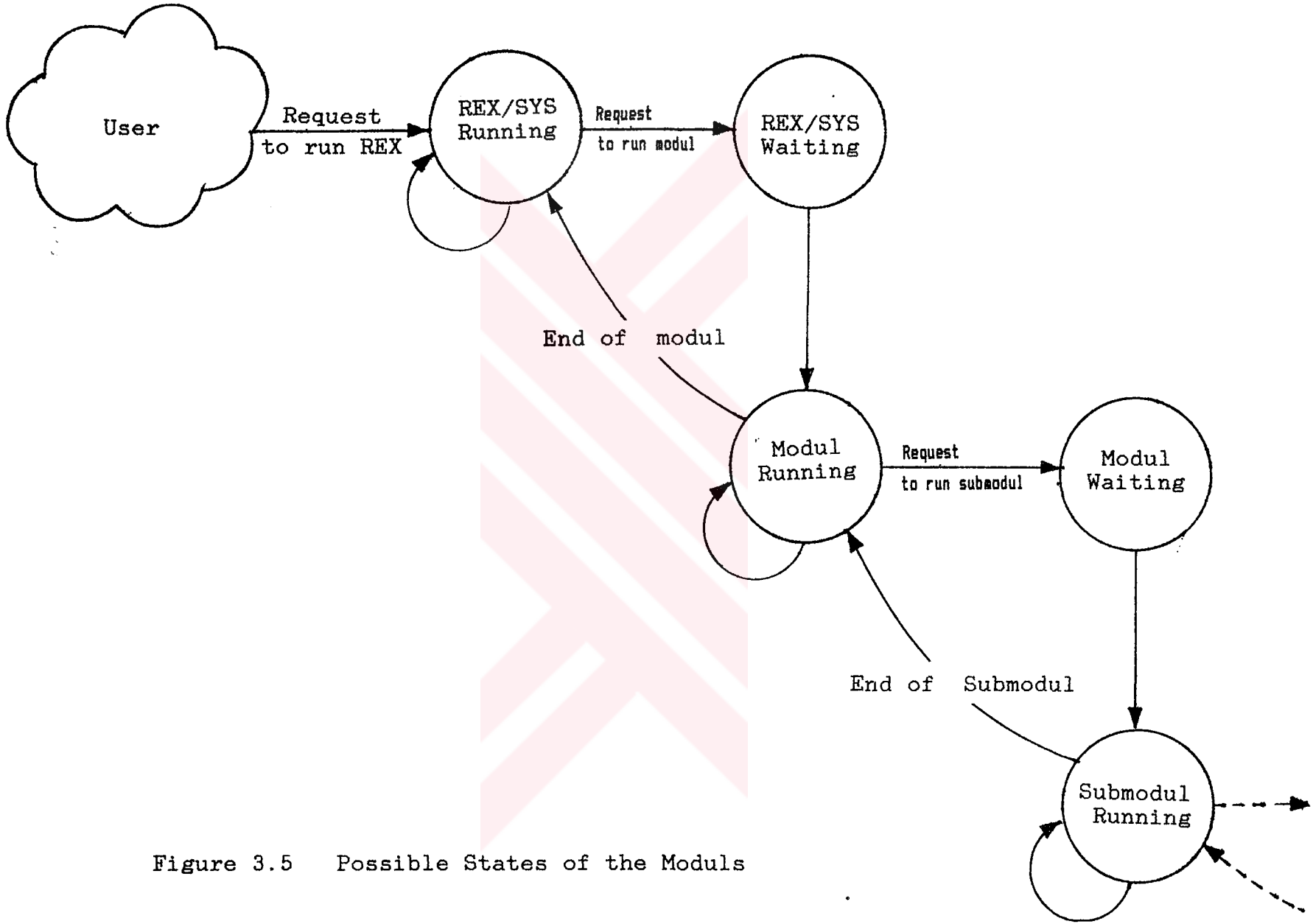


Figure 3.5 Possible States of the Moduls

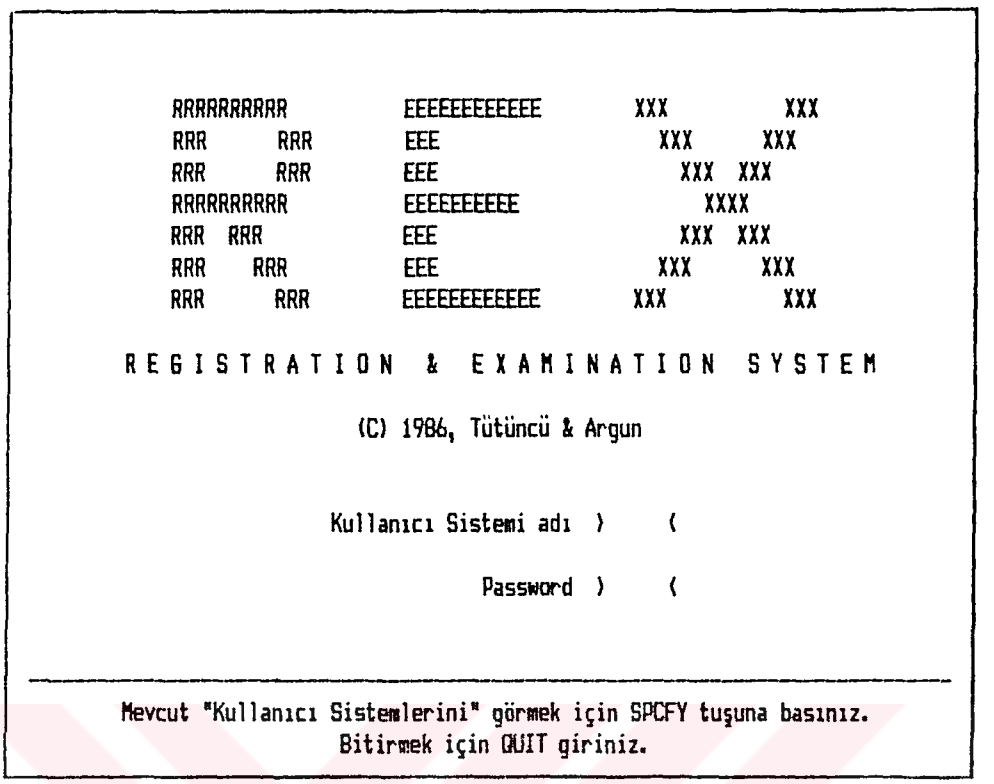


Figure 3.6 REX_Main Menu

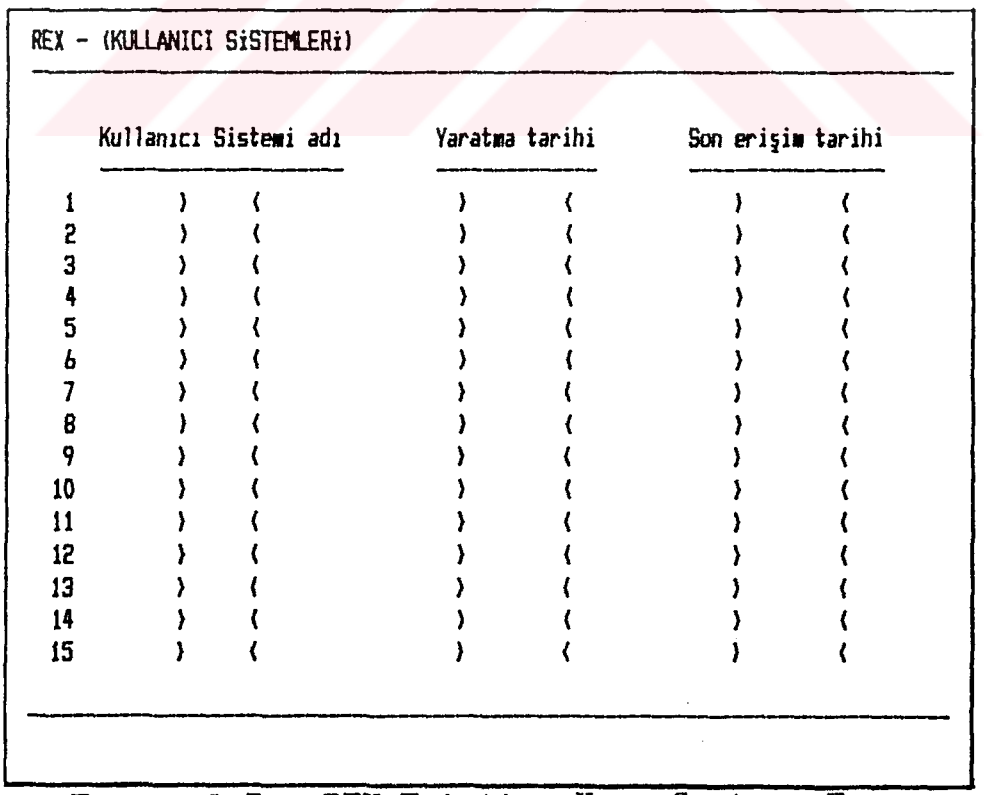


Figure 3.7 REX_Existing User Systems Form

If the <user system name> entered using the form/menu given in Figure 3.6 is not present in the system, a Parameter File is created using the form shown in Figure 3.8. After this stage, the menu shown in Figure 3.9 is used to activate one of the subsystems of REX and all the functions are executed in the chosen subsystem.

3.4.4.6. Properties of REX

-Some menus are used in order to give an idea about how to proceed on, to the users who do not know the characteristics of the software.

-It is assumed that, if some structures are formed in REX, these must be known by the users.

-REX is designed to be reentrant for multi user access.

-The subsystems of REX consist of more than one programs running synchronously.

-Special structures are used to obtain synchronization between programs.

-The users shall be able to stop working and continue from the point where they left working.

-REX can handle the examination for at most 100.000 persons.

-The examination can be performed in maximum 90 centers, 4 sessions and 10 areas in each sessions.

-Each area can be divided into 5 subareas.

-In every subarea, maximum 200 questions can be asked, however the total number of questions per area can not exceed 500.

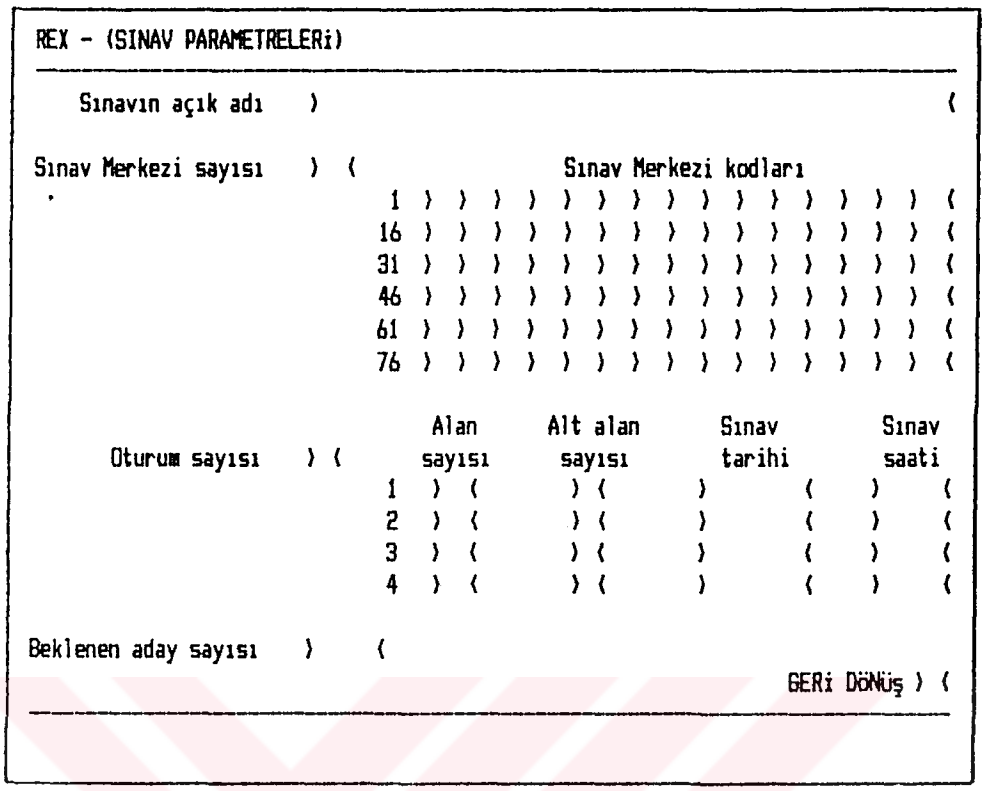


Figure 3.8 REX_User System Parameters Form

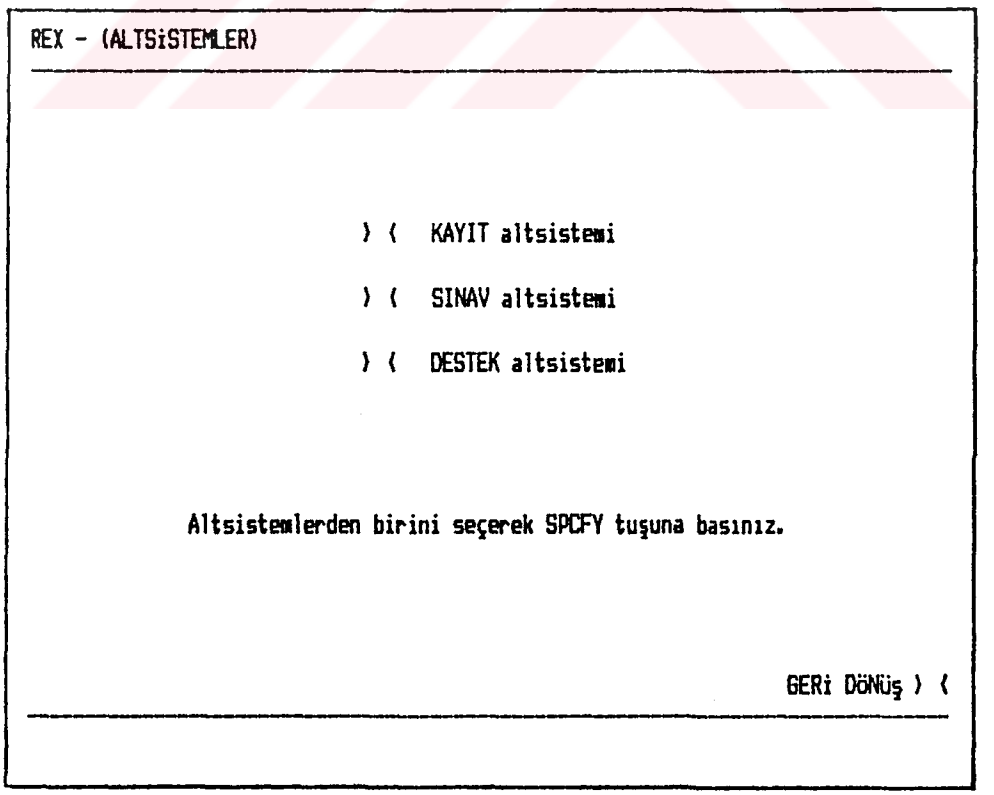


Figure 3.9 REX_Subsystems Form

-An individual score for each subarea and a success score obtained from the combination of subarea shall be calculated.

3.4.4.7. System Requirements of REX

-REX is designed for Burroughs Mainframe Systems.

-It can operate with operating systems like MCP 3.5 and later versions.

-The terminals to be used must be ET series or equivalent.

-The MCS to be used must be *SYSTEM/CANDE.

-The value of the LAISSEFILE, one of the operating parameters of *SYSTEM/CANDE, must be 6.

-A usercode as REX must be defined in the system. The password of this usercode must be REX, too.

-The following softwares need to be present in the system in order to activate the Supporting Subsystem:

*SMFII/LOGCONSOLIDATOR,
*SMFII/QUERY,
*SYSTEM/EDITOR,
*SYSTEM/DUMPALL

4. LOGICAL DESIGN OF THE EXAMINATION SYSTEM

As mentioned in chapter three, the Examination Organization System consists of four subsystems as **REX Subsystem, Examination Supporting Subsystem, Registration Supporting Subsystem and Test Preparation Subsystem.** The diagram showing the subsystems which form the Examination Organization System was given in Figure 3.2 of section 3. REX consists of three subsystems as **Registration, Examination and Supporting.** The related diagram was given in Figure 3.3 of the section 3. REX Examination Subsystem and Examination Supporting Subsystem which is a subsystem of the Examination Organization System form the Examination System. The subsystems which form the Examination System is shown in Figure 4.1 by using DFD and Fi representation has been used on the DFD while data flows were being coded. All data flows used in this chapter are given in Table 4.1.

A schema showing the relation between the Examination System and Examination Organization System, and the whole structure is given in Figure 4.2

REX Examination Subsystem starts operation by a process called "Room Assignment" which defines the examination rooms of the candidates. After the Room Assignment, it executes the "Printing Process" to print all documents to be used in the examination. It evaluates the results of candidates after the examination. An Examination Supporting Subsystem is designed for organizing the manpower to support following functions.

- Printing all documents
- Sending these documents
- Collecting these documents
- Controlling and if necessary sending back these documents
- Transferring the coming documents to REX

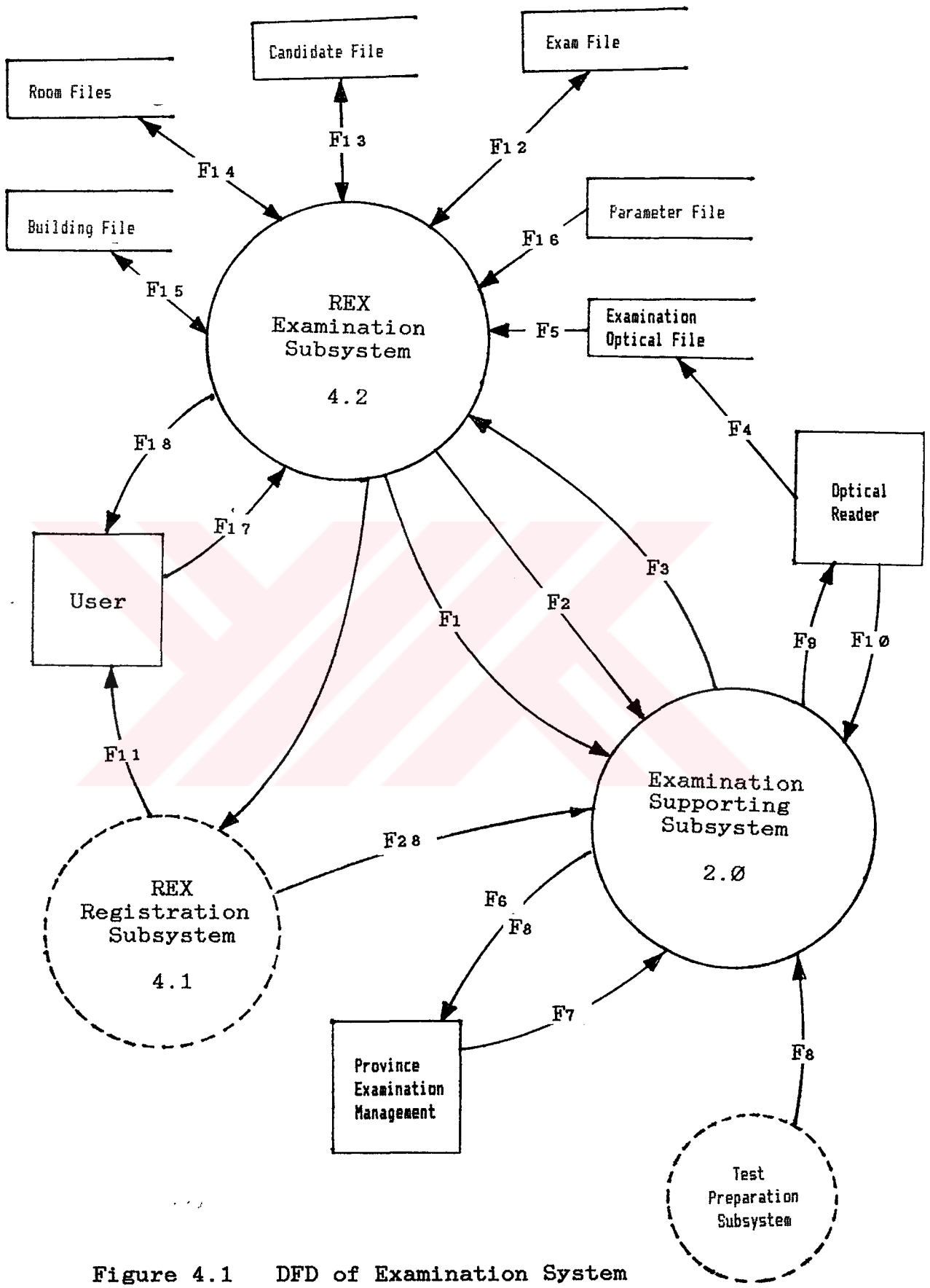


Figure 4.1 DFD of Examination System

Table 4.1 Data Flows

Fi	Definition
1	Archive Numbers of Unread Answer Sheets
2	Examination Documents
3	Examination Reports
4	Optically Read Answer Sheets
5	Candidate Examination Information
6	Examination Documents (Before Exam)
7	Examination Documents (After Exam)
8	Examination Question Booklets
9	Answer Sheets to be read
1Ø	Faulty Answer Sheets
11	Candidate Distribution
12	Exam File Information
13	Candidate File Information
14	Room File Information
15	Building File Information
16	Parameter File Informaton
17	User Command to REX
18	REX Reply to User
19	Temporary File Information
2Ø	Correctly Read Answer Sheets
21	Answer Keys
22	Avarages, SDs and Raw Scores
23	Standartd Scores
24	Evaluation Parameters
25	Outgoing Coefficients
26	Archive Coefficients
27	Assignment Data
28	Staff Info

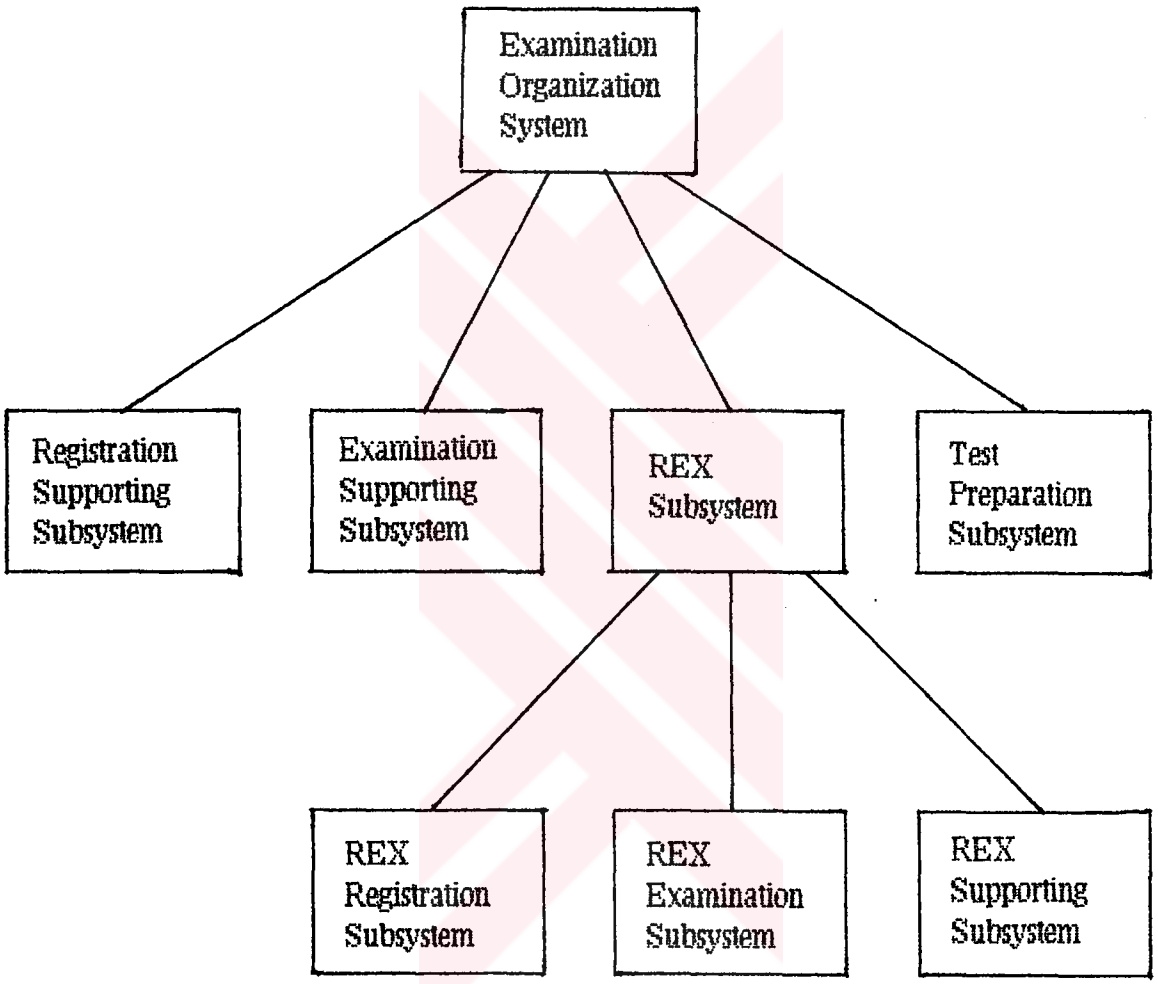


Figure 4.2 Examination Organization System Structure

4.1. REX Examination Subsystem

A diagram showing the general structure of this subsystem is given in Figure 4.3 by using DFD

The subsystem has three processes. These three processes are in close communication with the Examination Supporting Subsystem and Registration Subsystem which is one of the fundamental processes of the Examination Organization System. The three processes forming REX Examination Subsystem and their functions are as follows.

i) Room Assignment

This process selects the sufficient number of rooms for each session according to the candidates in the centers of the examination and places the candidates to these rooms. It also produces the "Answer Sheet Outgoing Numbers" and "Answer Sheet Archives numbers" which are required for the answer sheets to be used in the examination. It informs the Printing process about the all printouts to be performed after this process. Then, it informs the Examination Evaluation process to obtain the examination results and provides the continuity of the processes.

ii) Examination Evaluation

This is the process where the examination results are evaluated. In this process, answer keys of each subarea according to the sessions are obtained and by use of candidate answers from the Optical File, the correct and incorrect answers are calculated. A raw score is calculated for each subarea in each session. The incorrect answers effect the raw score according to the type of the examination. After the calculation of raw scores, averages and standard deviation for each subarea is calculated.

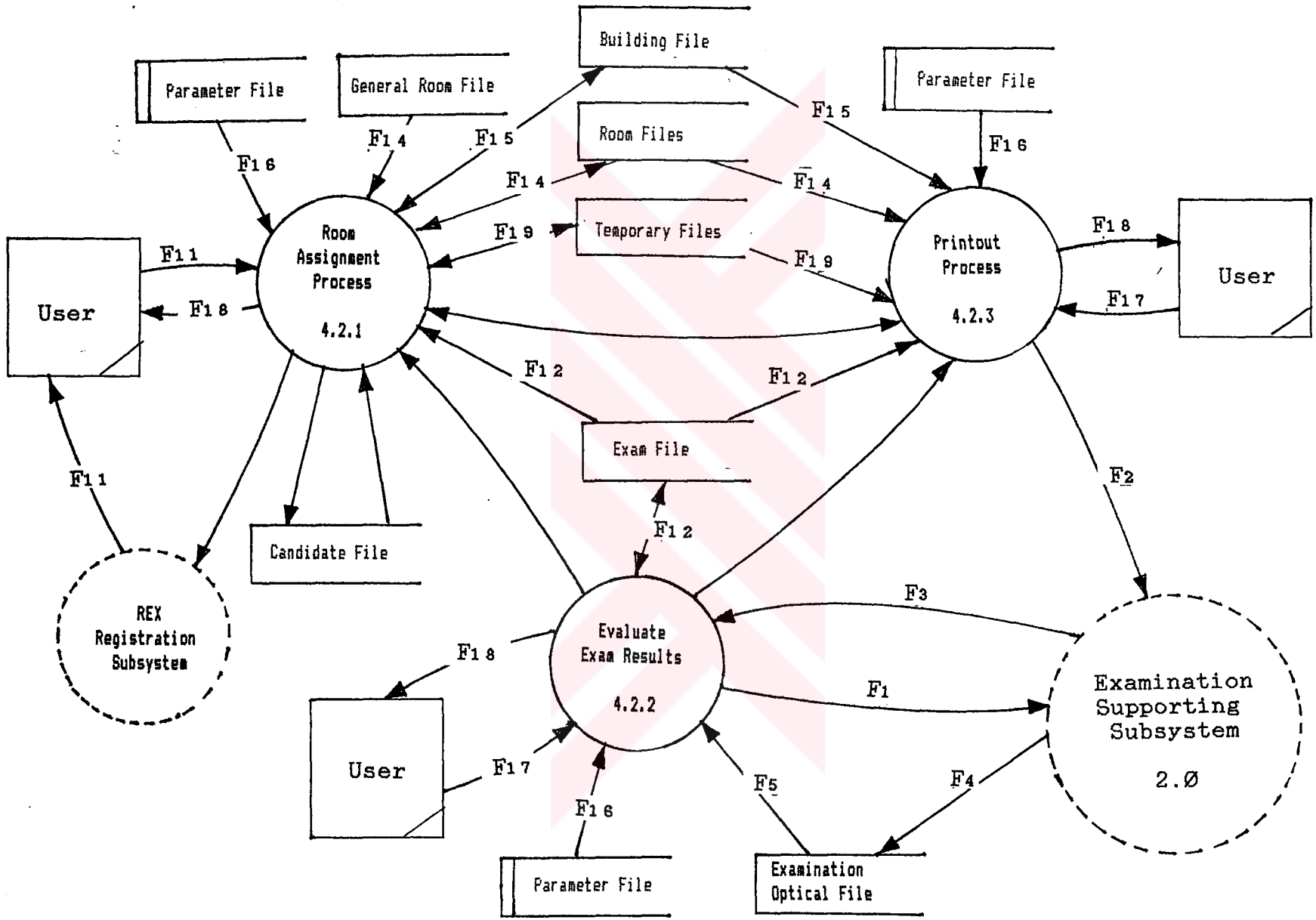


Figure 4.3 DFD of REX Examination Subsystem

By use of these averages, standard deviation, and raw scores, a standard score for each subarea is obtained. After all calculations, the weighted score for each candidate, in each session is calculated. Then, the processes dependent on the examination reports from the Province Examination Managements, is executed (e.g cancelation of cheating cases). Before the evaluation process comes to an end, the answer sheets which are not evaluated because of their incomplete optical readings are identified and evaluated. As a last step, the Registration System is informed in order to announce the results to the candidates.

iii) Printing

This process executes all printing processes related with the examination, after the Room Assignment. These printouts are as follows.

-Answer Sheet
-Examination Document Delivery List
-Room Identification Card
-Room Candidate Attendance List
-Room Examination Report
-Extra Document Envelope
-Building Extra Document Utilization Schedule
-Rooms List
-Examination Document Receiving List

4.2. Examination Supporting Subsystem

This subsystem is fundamentally designed to provide supporting services for the REX Examination Subsystem. During the operation of REX Examination Subsystem, most of the input/output information of this subsystem is obtained by series of processes using manpower. All processes, such as the control of information and printouts produced by the REX Examination Subsystem make it necessary to construct an Examination Supporting

Subsystem operating in this manner. The operation of this subsystem is shown in Figure 4.4 by using DFD. The processes forming the Examination Supporting Subsystem consists of four activities as; Document Mailing, Document Collecting, Document Control and Correcting, Archiving. The detailed information and diagrams of these processes will be introduced in the following sections.

4.3. Detailed Logical Design of REX Examination Subsystem

It is necessary to consider in detail the Room Assignment, Examination Evaluation and the Printing processes which are the three fundamental processes of REX Examination Subsystem of which its overview diagram was given in Figure 4.3.

4.3.1. Detailed Logical Design of the Room Assignment Process

Room Assignment is the process performed to identify the examination places of candidates in each session. Details are shown in Figure 4.5 by using DFD. The processes are executed as follows.

Before the assignment process, candidate distribution defined by the Registration System is obtained by using a form on the screen. A Distribution File is created in order to provide for the later uses of this distribution. In accordance with this file, by use of the information on the Parameter File, Room Files which contains sufficient number of rooms for each session are created. Also a single Building File which includes the buildings containing the rooms is created. After this stage, by using the Candidate File, the candidates who will take the examination in each session are chosen and Temporary Files are formed to make Room Assignment.

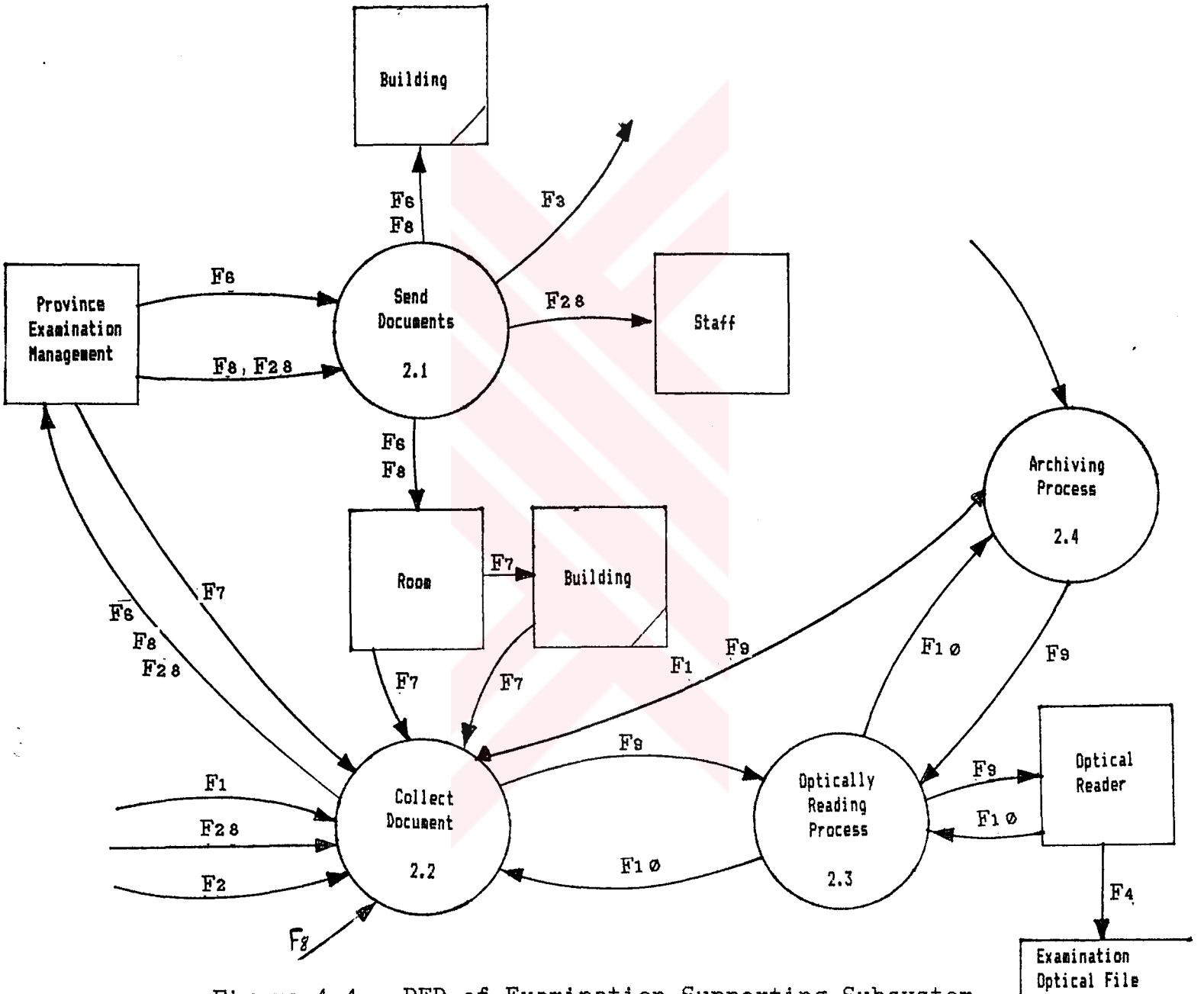


Figure 4.4 DFD of Examination Supporting Subsystem

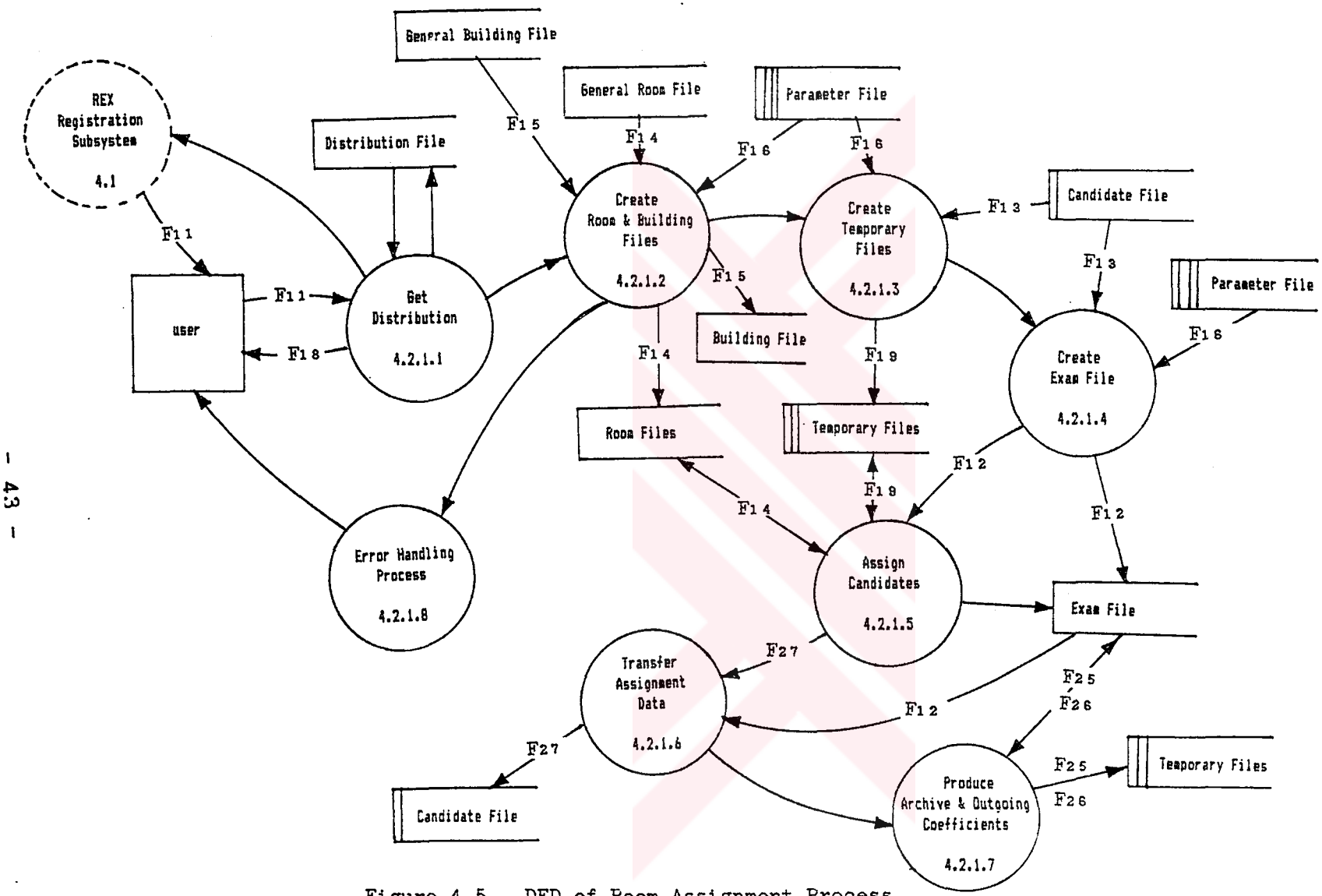


Figure 4.5 DFD of Room Assignment Process

After creating Room and Temporary Files, again by using the Candidate File and Parameter File an Examination File is created. At this stage, all necessary files for assignment have been created. Before the assignment process, these files are sorted. By use of sorted Room and Temporary Files of each session, the candidates are assigned to the rooms.

After this, assignment information is transferred to the Candidate File. The most important part of the Room Assignment process is the section called the "Placement of Candidates to the Rooms", so this section is shown in Figure 4.6 in detail. The function of this section is as follows.

The Room Files are sorted according to the area code in the examination center, assignment priority in the area code, room name in the assignment priority. Temporary Files are sorted according to the area code in the examination center, random number in the area code. After this, the candidates are assigned to the rooms and this information is recorded on the Examination File. When the assignment process comes to an end, all assignment information is transferred from the Examination File to the Candidate File.

Another process has been designed to generate the archive coefficients and transfers these coefficients first to the Temporary files and then to the Examination file, for the archiving of the answer sheets. The details of this process are shown in Figure 4.7 by using DFD. In this process, first the Temporary files are sorted according to the application number of the candidates in the examination center. Therefore, the candidates of each center are divided in groups of 1000 persons and to each group, a box number starting from 1 is assigned.

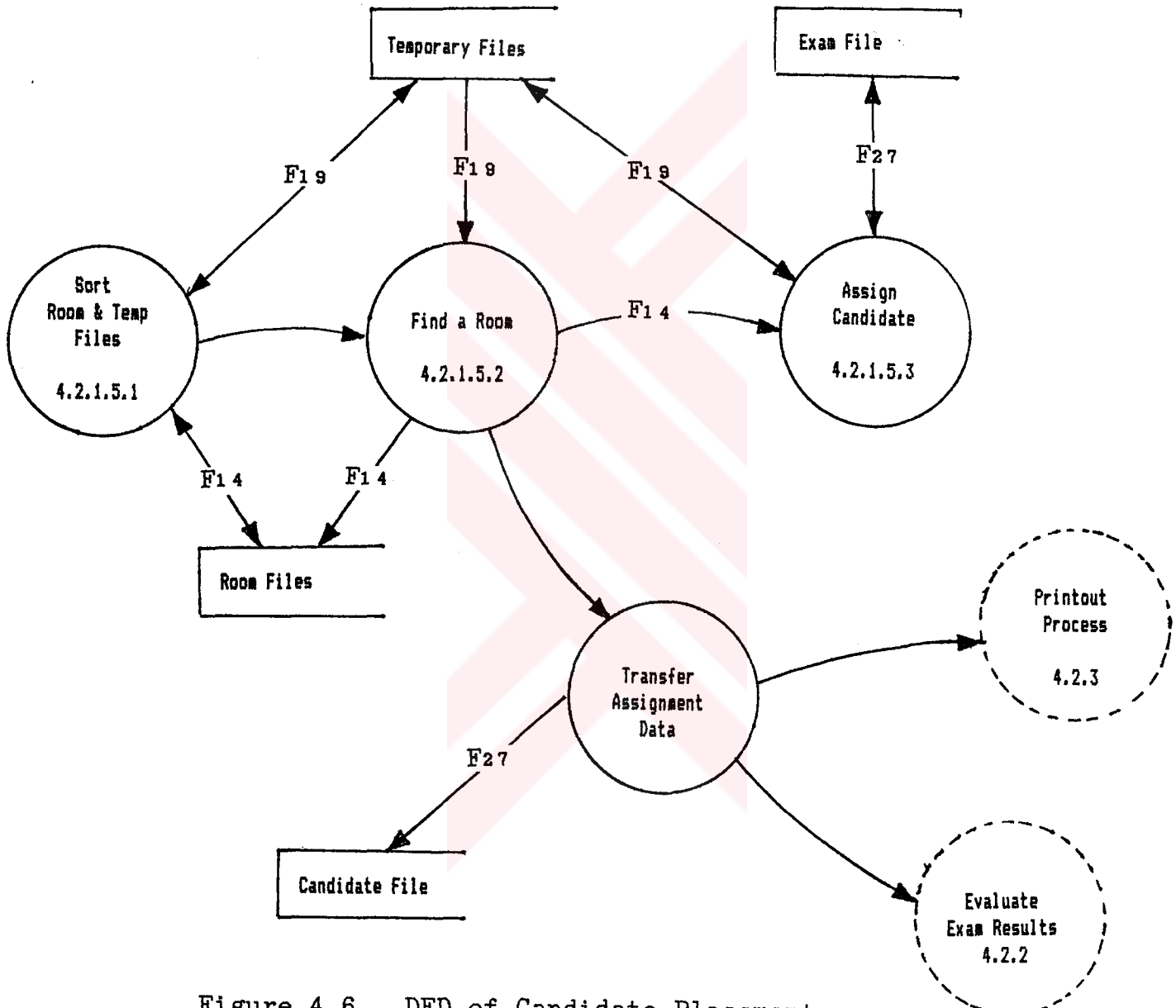


Figure 4.6 DFD of Candidate Placement

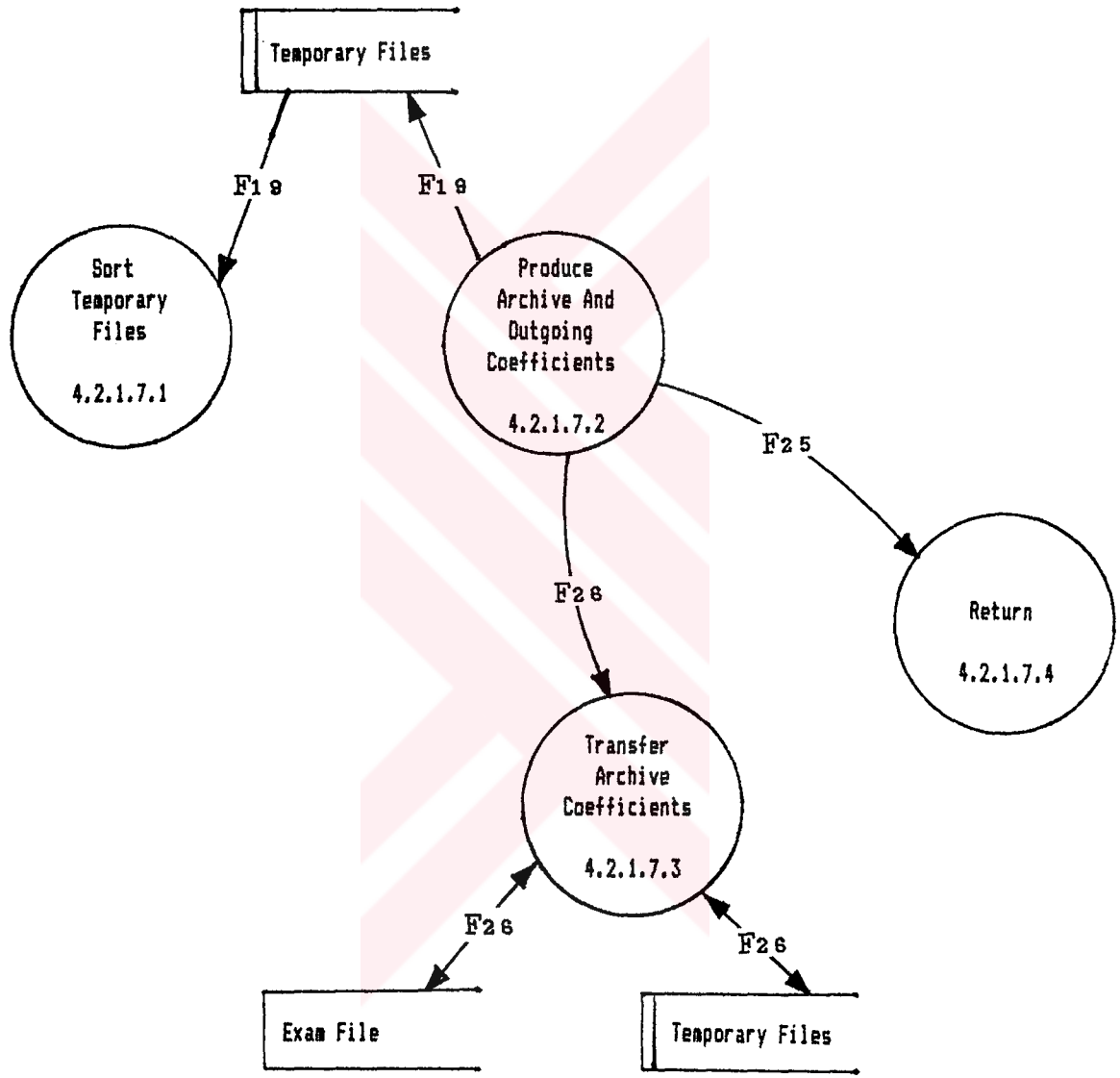


Figure 4.7 DFD of Processes Related to Archiving

This process goes on till the last candidate. The same procedure is applied on the other centers. The archive numbers for a center will be as follows.

Ø1ØØ1, Ø1ØØ2,...Ø1999
Ø2ØØ1, Ø2ØØ2,...Ø2999
.
.
.
nnØØ1, nnØØ2,...nn999

The archiving process is repeated for the candidates of each session. When the archiving process is completely terminated, these numbers are transferred to the Examination File from the Temporary Files.

After that, another process which generates the "Answer Sheet Outgoing Coefficients" and which transfer them to the Temporary File, that will be used to send the answer sheets to the examination centers, comes into action. The details of this process are also given in figure 4.7.

In this process, first the Temporary Files are sorted according to the examination center, building number, room number and application number. Therefore, a number starting from 1 is assigned to the candidates within the building for each center. When a new building is included, the sequence of the number assignment starts again from 1.

This number assignment process goes on till the last building in a center. The some procedure is applied on the other centers. The outgoing coefficients are generated separately for each candidates of each session. These coefficients will be present only in the Temporary Files.

4.3.2. Detailed Logical Design of the Examination Evaluation Process

The examination evaluation process evaluates the answers of the candidates taking the subareas within each area for each session and calculates the corresponding scores. These scores are a standard score for each subarea and a weighted score for each session. The details of this process are shown in Figure 4.8 by using DFD.

The answer keys are taken for the screen by using a form from the screen for each subarea in the examination. These answer keys will be recorded to the Answer Keys File considering that they might be used again in case of an interruption of the process. For this process, the Parameter File is used as well. The information is transferred to the "Raw Scores Calculation Process" in order to enable the calculation of the raw scores. Within this process the candidates' answers from the Optical File and the answer keys are used together and thus the raw scores are obtained.

During the raw score calculation, the effect of the incorrect answers are determined by the information supplied by the user of the software. At the end of this process all the average and standard deviation values of the whole subareas, are being calculated. Using the average, standard deviation and raw scores of each subareas a standard score is calculated. The process which calculates this score, uses the information recorded in the Parameter File and records the obtained scores to the Examination File. Then, using the standard scores of all the subareas, a weighted score is calculated for each session. Before calculating these scores, the weights of the standard scores to be used, are taken from the user by menus.

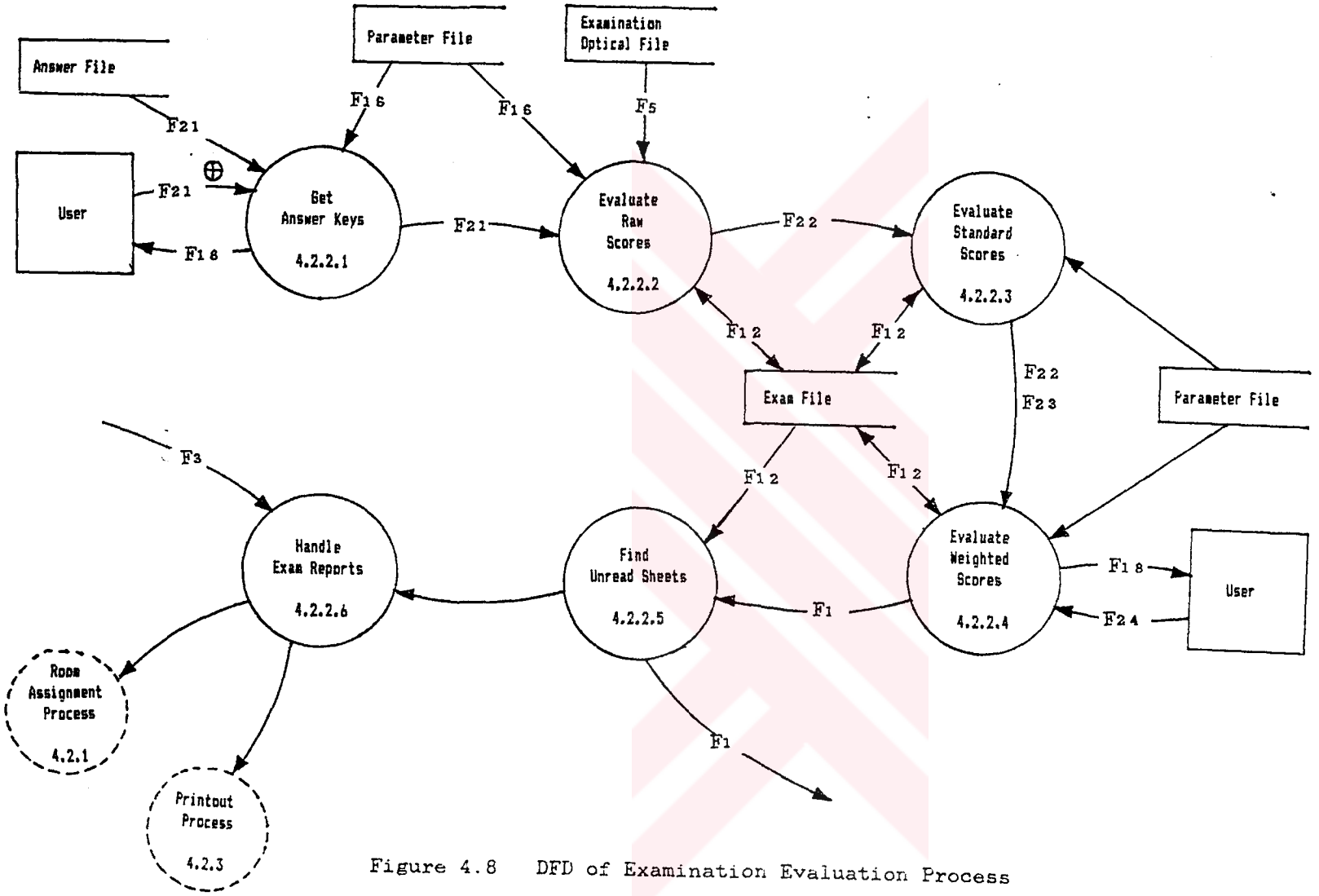


Figure 4.8 DFD of Examination Evaluation Process

After the calculation of the weighted scores, they will be transferred again to the Examination File. At this time, the evaluation process in said to be completed. However, there are still some answer sheets which are not evaluated because of various reasons. In order to obtain and ensure the evaluation of these sheets, the necessary archive information are retrieved and transferred to the Examination Supporting Subsystem. Finally the information like cheating or breaking the examination rules etc. recorded during the examination, are transferred to the Examination File.

4.3.3. Detailed Logical Design of the Printing Process.

The Printing process contains many printouts related to the examination preparations and the delivery of the documents, activities during and after the examination deliveries. These processes are shown in Figure 4.9 by using DFD. All printing processes are executed and then all the printouts sent to the Examination Supporting Subsystem. All printouts and their descriptions is as follows.

- Answer Sheet: These are the answer sheets to be used during the examination. Answer sheets are printed according to the names of the candidates of each session. The printing is executed according to the Answer Sheet Outgoing Coefficients. These coefficients are printed on the answer sheets. The answer sheets coming after the examination are received and archived according to the Answer sheet Archive Number. The archive numbers are also printed on the answer sheets.

- Examination Documents Delivery List: This is the list to be used during the packing process of all examination documents to be mailed to the examination centers.

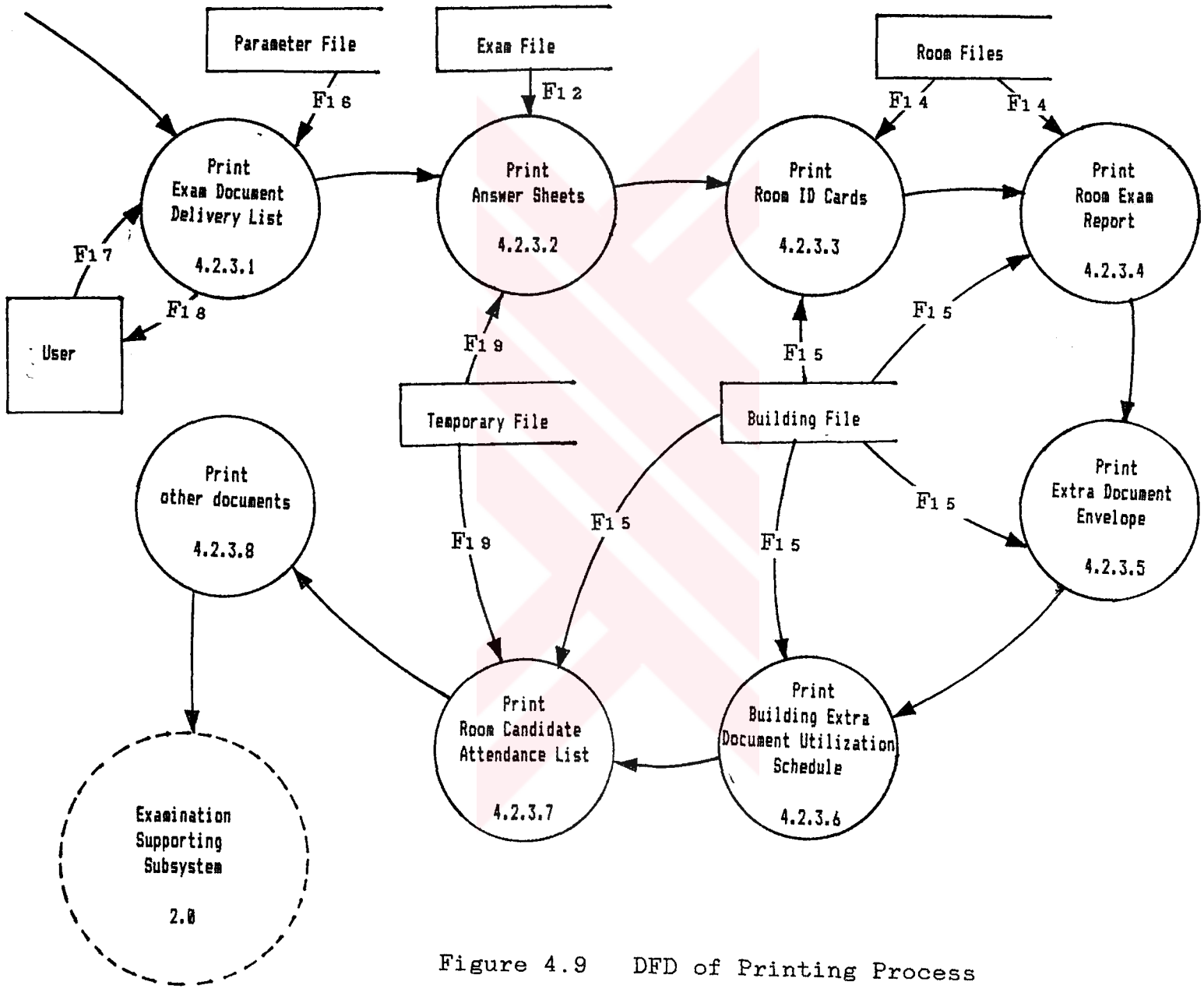


Figure 4.9 DFD of Printing Process

- Room Identification Card: It is a card which identifies the examination rooms and which is placed on the room doors.

- Room Candidate Attendance List: It is the list which includes the candidates to take the examination. It is used to check attendance during the examination.

- Room Examination Report: The report to be used by the Room Director in the examination rooms.

- Extra Document Envelope: The envelopes to be used when the real examination documents are not complete. They are arranged according to the buildings in the examination centers.

- Building Extra Documents Utilization Schedule: The schedule on which the reports are recorded when the extra examination documents are used.

- Rooms List: The list which contains the rooms of the buildings in the examination centers.

- Examination Documents Receiving List: The list which shows the documents coming after the examination.

4.4. Detailed Logical Design of the Examination Supporting Subsystem.

The details of the Examination Supporting Subsystem general schema of which was given in Figure 4.4 is shown in Figure 4.1Ø by using DFD.

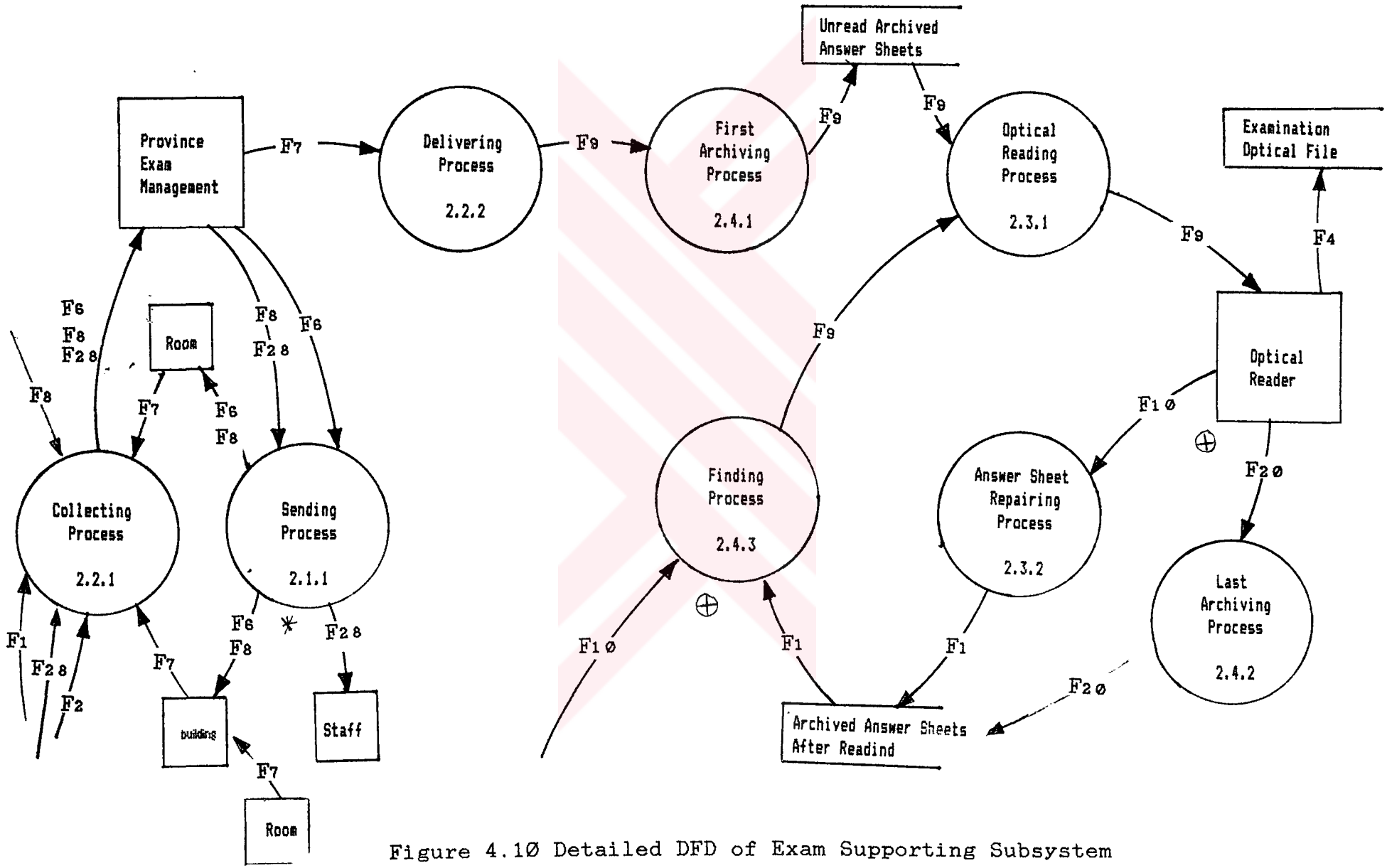


Figure 4.1Ø Detailed DFD of Exam Supporting Subsystem

4.4.1. The Purpose of the Examination Supporting Subsystem

The purpose of this subsystem is to provide supporting services for REX Examination Subsystem. Most of these services are dependent on manpower.

The computer controlled systems like optical readers and machines like output separators and cutters are also in scope of these services. Besides these, archiving of all the Answer Sheets used in the examination and finding these when required is also one of the services of this system.

4.4.2. Activities of the Examination Supporting Subsystem

Fundamentally, there are four main processes of this Subsystem.

- Document Sending Process: All duties performed to send all documents coming from any subsystem to the rooms or buildings in the examination center is executed by this process.

- Document Collecting Process: The process in which the examination documents are received and counted after the examination. The documents are counted one by one and the checked ones are sent to the Document Control and Correction process.

-Document Control and Correction Process: The process in which the documents obtained by Document Collection Process is checked and optically read, and corrected. If required, contacts with the Archiving Process can be provided and with the help of the information from this, the required answer sheets can be supplied.

-Archiving Process: The process in which the answer sheets are archived and provided when necessary.

The new processes and their definitions, which are formed by defining four processes above are as follows.

- Distribution Process: Delivers the examination documents from Examination Supporting Subsystem to the Province Examination Management first, and then from there to the buildings or rooms and the examination staff.

- Collection Process: The examination documents are collected from buildings and rooms and then sent to the Examination Supporting Subsystem.

- Receiving Process: Receives the examination documents obtained by the collection process from the Examination Supporting Subsystem by performing the necessary controls and counts. Transfers the correct examination documents to the Primary Archiving Process and corrects the errors of the documents.

- Primary Archiving Process: By using archive numbers, it archives the Answer Sheets coming from the Receiving Process prior to the Optical Reading Process.

- Optical Reading Process: The unread or corrected Answer Sheets are fed to the optical readers. The answer sheets read with no errors are sent to the Final Archiving Process and the information on them are recorded to the Optical File. The unread sheets are transferred to the Answer Sheet Reparation process.

- Answer Sheet Reparation Process: This process makes readable the answer sheets which are damaged during the Optical Reading Process or the ones which are not readable because of faulty prints and sends them to the

Optical Reading Process, to be read again.

- Archive Retrieving Process: This process, finds the related sheets using the archive numbers concerning the unread answer sheets and sends them to the Optical Reading Process in order to be read again.

- Final Archiving Process: This process enables the answer sheets, which are correctly read via optical reading process, to be archived by using the archive numbers.

In this subsystem, the processes in the newly defined sections are executed as follows.

-By the use of the Distribution Process, documents related with the staff from the Registration System, Room and Building documents from the Examination Supporting Subsystem, Test documents from the Test Preparation Subsystem are delivered first to the Province Examination Management and then to the rooms, buildings and staff.

-After the examination, the examination documents are collected in the Province Examination Management by means of the Collection Process.

-The collected examination documents are taken from the Province Examination Management by using the Receiving Process. These documents are checked and the answer sheets are transferred to the Primary Archiving Process in order to be archived.

-The answer sheets which are received and archived, are fed to the optical reader. The answer sheets which are read with no error are sent to the Final Archiving Process and the information on these sheets form the Optical File.

-The sheets which are damaged or read with errors are sent to the Answer Sheet Reparation Process. The Answer Sheet Reparation Process, corrects the answer sheets and transfers them to the Optical Reading Process again.

-After all readings, evaluation of the information on the Optical File, by REX Examination Subsystem, the archive information about the unread sheets are sent to the Archive Retrieving Process. The Archive Retrieving Process provides the related sheets, by using the archive numbers. These sheets are transferred to the Optical Reading Process in order to be read again.

-This process continues till no unread answer sheets are left. After the examination, as well as these activities related with the examination documents, reports related to the entire examination comes from this process evaluated by REX Examination Subsystem.

5. PHYSICAL DESIGN OF THE EXAMINATION SYSTEM

In this section, all the file structures, record layouts of these files, used programs, special files which are used in the design of REX Examination Subsystem, and their purposes, are described.

5.1. Data Files and Record Layouts

In this section, all data files used in this subsystem are described. The record layouts of each data file is given by using Warnier/Orr diagram and the physical structure of the related data fields are also described in this diagram. When describing the physical structure of the data fields, the abbreviations CHAR and NUM are used for alphanumeric and numeric fields, respectively. In order to define the length of the data field, an integer or rational number is used after the abbreviation. The integer part of the given number defines the integer part length and the number after the rational part defines the rational part length. All lengths are given in byte units.

As an example, lets examine the following Warnier/Orr diagram.

```
        X [CHAR 23]
XYZ     Y [NUM 6]
        Z [NUM 5.2]
```

Here, the XYZ data field is defined by Warnier/Orr diagram. XYZ data field consists of three data, X,Y and Z. X; 23 bytes long alphanumeric, Y; 6 bytes long numeric and Z; data having two digits of rational, five digits of integer numbers. In addition to the Warnier/Orr representation, just after the diagram, another figure is given to show real variable names which will be used in the software.

i) Candidate File

It is the file where the information about the candidates are stored. It is created and updated by the Registration System. The data fields of the records forming this file and their physical names are shown in Figure 5.1 by using Warnier/Orr diagram and Figure 5.2 respectively. The definition of data fields is as follows.

-Application Number: It is the unique number given to the candidates. This number is also used to access the file.

-Name, Surname, Father's Name, Date of Birth, Sex: These are the personal information about the candidate.

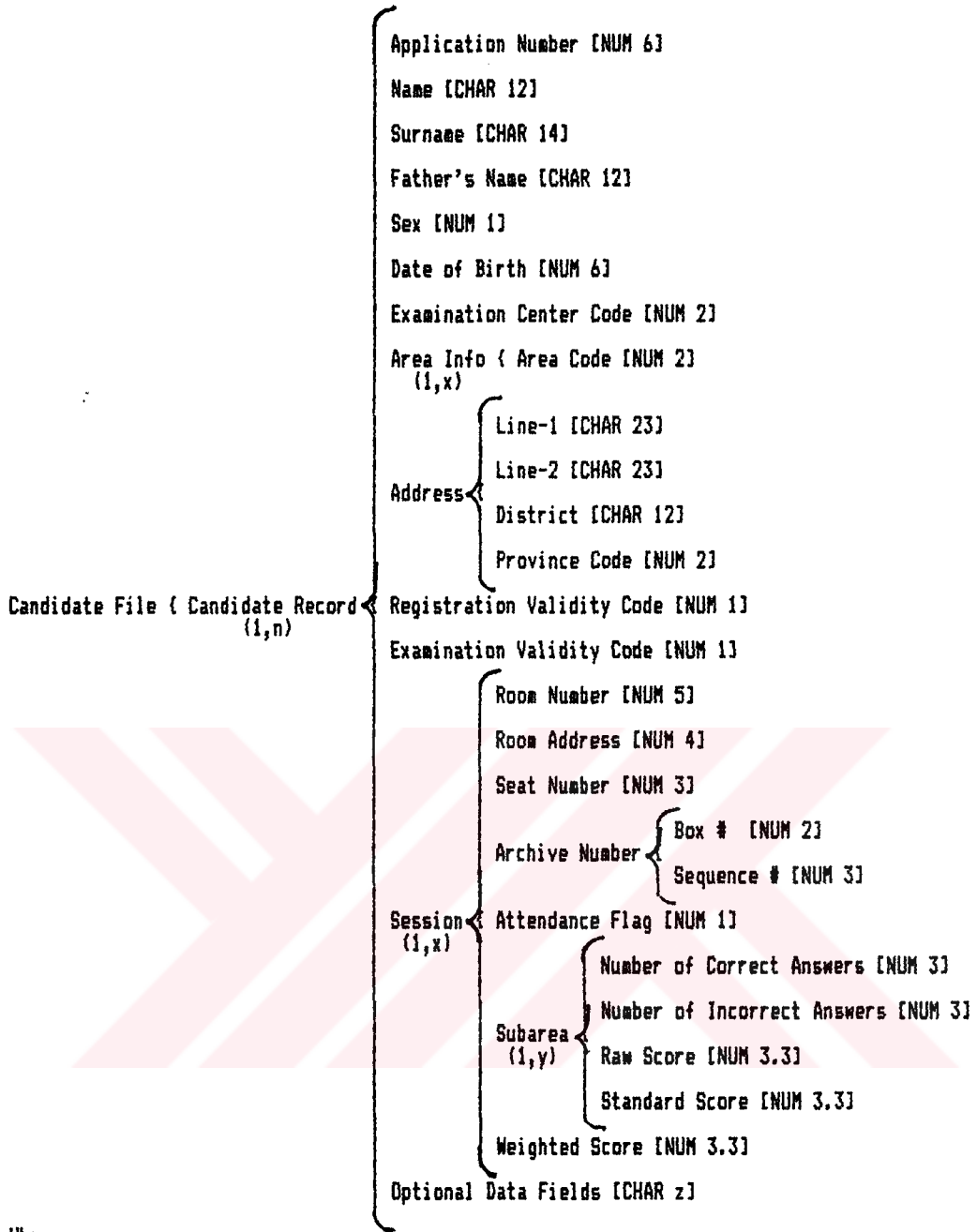
-Examination Center Code: It is the code which identifies the examination center that the candidate wishes to take the examination. If the center is a province, then a province traffic code is used. If not, the numbers larger than 67 are used. It is used by REX Examination Subsystem during the Room Assignment process.

-Area Code: It defines the area that the candidate is going to take the examination in each session. For each session, the area codes start from 1 and continues up to the number of areas. It is used in Room Assignment and Examination Evaluation processes.

-Address: Mailing address of the candidate.

-Registration Validity Code: It indicates whether the application of the candidate is valid or canceled because of the insufficient conditions.

-Examination Validity Code: It indicates whether the examination of the candidate is valid or canceled because of breaking the examination rules.



Where,

n: is the number of the candidates.
x: is the number of the sessions for the examination.
y: is the number of subareas for each session.
z: is the length of optional data fields

Figure 5.1 Warnier-Orr Diagram of Candidate File Layout

```
DCL 1 CANDIDATE_REC          CONTROLLED,
      2 APPLICATION#         PIC'999999',
      2 NAME                 CHAR(12),
      2 SURNAME              CHAR(14),
      2 FATHER               CHAR(12),
      2 SEX                  PIC'9',
      2 BIRTH                PIC'999999',
      2 CENTER               PIC'99',
      2 AREA(#OFSESSIONS)    PIC'99',
      2 ADDRESS,
        3 LINE1              CHAR(23),
        3 LINE2              CHAR(23),
        3 DISTRICT           CHAR(12),
        3 PROVINCE           PIC'99',
     2 REG_VALIDITY          PIC'9',
     2 EXAM_VALIDITY         PIC'9',
     2 SESSION(#OFSESSIONS),
        3 ROOM#              PIC'99999',
        3 ROOM@              PIC'9999',
        3 SEAT#              PIC'999',
        3 ARCHIVE#,
          4 BOX#             PIC'99',
          4 BOXSEQ#          PIC'999',
        3 ATTENDANCE         PIC'9',
        3 SUBAREA(MAX#OFSUBAREAS),
          4 CORRECT          PIC'999',
          4 INCORRECT        PIC'999',
          4 RAW              PIC'-999.V999',
          4 STANDARD         PIC'-999.V999',
       3 WEIGHTED            PIC'-999.V999',
    2 OPTIONAL               CHAR(SIZEOFOPTIONAL);
```

Figure 5.2 Variable Names of Candidate File Layout

-Room Number: It indicates the room in which the candidate will take the examination. This number is unique for each room. It is used in the room Assignment Process.

-Room Address: It is the address of the room in the Room File of the related session.

-Seat Number: It indicates the seat number on which the candidate will take the examination in the room. The Seat Number starts from 1 and continues up to the capacity of the examination room.

-Archive Number: It is the number given to archive the answer sheets after the examination. This number, together with the the examination center code, helps to find the answer sheets easily when required moment. The archive number consists of box# and box_sequence#. The sheets can be easily accessed via box# in the provinces and box_sequence# in the boxes. The box# changes between 1 and 99, and the box_sequence# changes between 1 and 999. The box# starts from 1 for each province.

-Attendance Flag: It indicates whether the candidate has attended the examination or not.

-Number of Correct Answers: It is the number of correct answers given by the candidate for each subarea.

-Number of Incorrect Answers: It is the number of incorrect answers given by the candidate for each subarea.

-Raw Score: It is obtained by subtracting a certain percentage of incorrect answers from the number of correct answers. This percentage is 1/4 in most of the existing applications.

-Standard Score: It is obtained by the transformation of raw score to the scores having a mean of 5Ø and standard deviation 1Ø.

-Weighted Score: It is obtained by multiplicating the standard scores of each subarea with a definite coefficients and their summation.

-Optional Data Fields: In some examinations, it might be necessary to have some data fields for purposes other then the ones for the predefined data fields. For this purpose, the user may define new data fields in the Candidate File. These fields are completely under the control of the user and can be named in any form.

ii) Examination File

This file is created by selecting candidates to take the examination after the completion of the registration process. The evaluation of examination and determination of scores are executed via this file. The scores obtained after the evaluation are transferred to the Candidate File from this file. The data fields of the records forming this file and their physical names are shown in Figure 5.3 by using Warnier/Orr diagram and Figure 5.4 respectively. The description of the data fields is as follows.

-Application Number: It is the unique number given to the candidates. This number is also used to access the file. After the completion of the Room Assignment and Examination Evaluation processes, using this number as a key, all the examination from the Examination File are transferred to the Candidate File.

-Room Number: It indicates the room in which the candidate will take the examination. This number is unique for each room. It is used in the Room Assignment process.

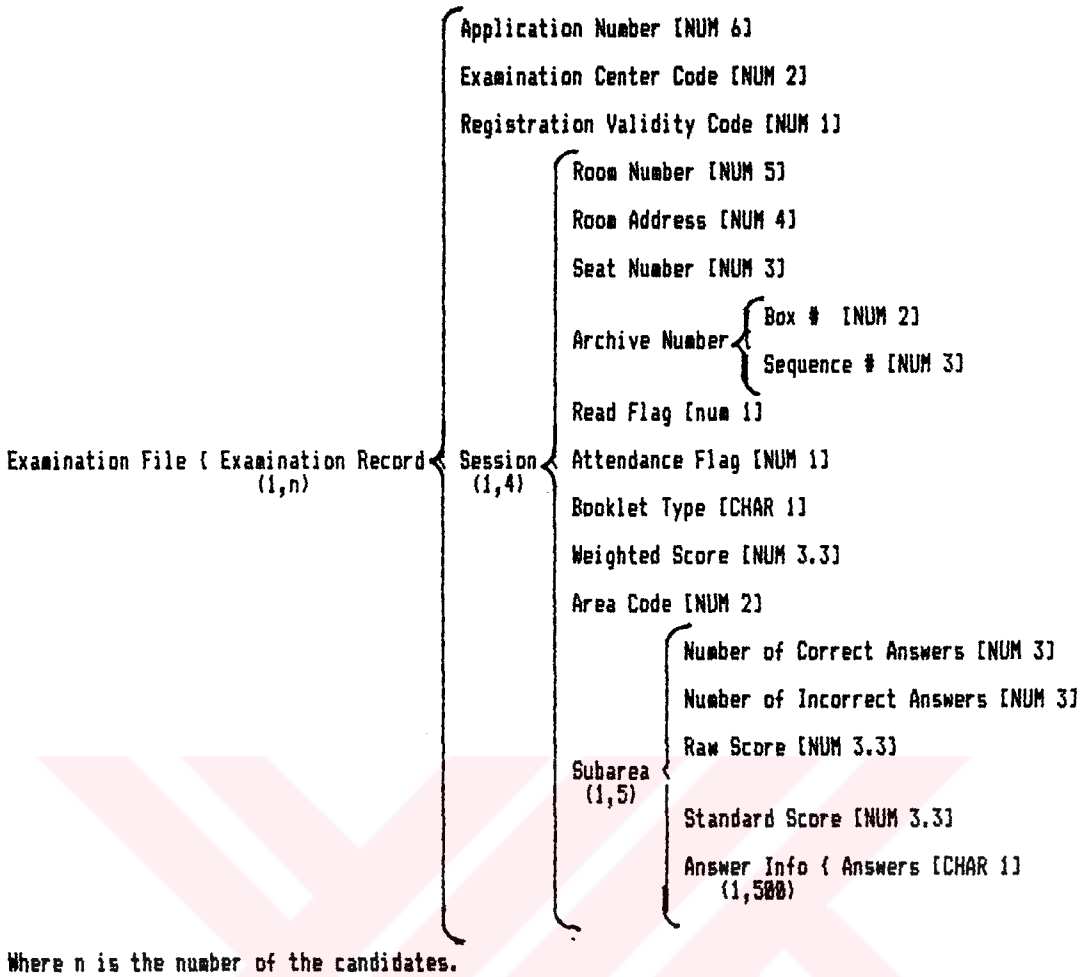


Figure 5.3 Warnier-Orr Diagram of Exam File Layout

```
DCL 1 EXAM_REC,
      2 APPLICATION#          PIC'999999',
      2 CENTER                PIC'99',
      2 SESSION(4),
        3 ROOM#               PIC'99999',
        3 ROOM@               PIC'9999',
        3 SEAT#               PIC'999',
        3 ARCHIVE#,
          4 BOX#              PIC'99',
          4 BOXSEQ#           PIC'999',
        3 READ_FLAG           PIC'9',
        3 ATTENDANCE          PIC'9',
        3 BOOKLET_TYPE        CHAR(1),
        3 WEIGHTED            PIC'-999.V999',
        3 AREA                PIC'99',
        3 SUBAREA(5),
          4 CORRECT           PIC'999',
          4 INCORRECT         PIC'999',
          4 RAW               PIC'-999.V999',
          4 STANDARD          PIC'-999.V999',
          4 ANSWERS(200)      CHAR(1);
```

Figure 5.4 Variable Names of Exam File Layout

-Room Address: It is the address of the room in the Room File of the related session.

-Seat Number: It indicates the seat number on which the candidate will take the examination in the room. The Seat Number starts from 1 and continues up to the capacity of the examination room.

-Archive Number: It is the number given to archive the answer sheets after the examination. This number, together with the the examination center code, helps to find the answer sheets easily when required. The archive number consists of box# and box_sequence#. The sheets can be easily accessed via box# in the provinces and box_sequence# in the boxes. The box# changes between 1 and 99, and the box_sequence# changes between 1 and 999. The box# starts from 1 for each province.

-Reading Flag: It indicates that the answer sheets of the candidates are whether read or not. The papers which have not been read optically are identified by the help of this flag.

-Attendance Flag: It indicates whether the candidate has attended the examination or not.

-Booklet Type: In some examinations, different question booklets are prepared in order to prevent cheating. This indicator is used to indicate the question booklet type of the candidate.

-Number of Correct Answers: It is the number of correct answers given by the candidate for each subarea.

-Number of Incorrect Answers: It is the number of incorrect answers given by the candidate for each subarea.

-Raw Score: It is obtained by subtracting a certain percentage of incorrect answers from the number of correct answers. This percentage is 1/4 for most of the existing applications.

-Standard Score: It is obtained by the transformation of raw score to the scores having a mean of 5Ø and standard deviation 1Ø.

-Weighted Score: It is obtained by multiplicating the standard scores of each subarea with a definite coefficients and their summation.

-Answers: The answers given by the candidate for each subarea of the related session.

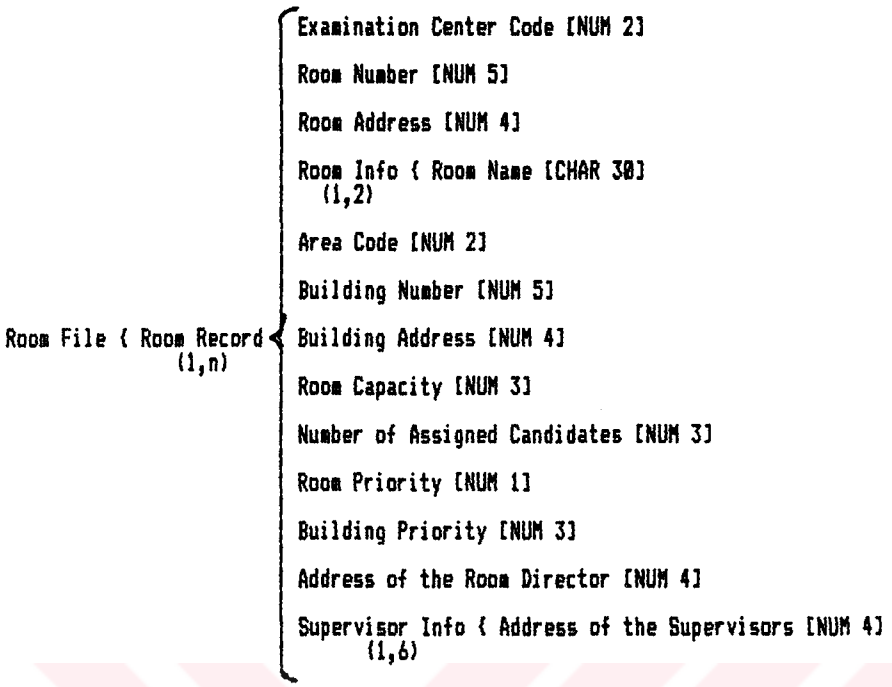
iii) Room Files

These are the files created by selecting sufficient number of rooms from the General Room File for each session. For each session of the examination, there is one Room File. During room assignment, it is used synchronously with Temporary Files. The data fields of the records forming this file and their physical names are shown in Figure 5.5 by using Warnier/Orr diagram and Figure 5.6 respectively.The description of the data fields is as follows.

-Room Number: It is the unique number defining the room. This number is recorded on the Examination File for the candidates assigned to this room during the room assignment process.

-Room Name: Full name of the room.

-Examination Center Code: It indicates examination center of the room. If the center is a province then the Province Traffic Codes are used. For other centers, the codes larger than 67 are used.



Where n is the number of rooms in the related session.

Figure 5.5 Warnier-Orr Diagram of Room File Layout

```
DCL 1 ROOM_REC,
      2 CENTER              PIC'99',
      2 ROOM#               PIC'99999',
      2 ROOM@               PIC'9999',
      2 ROOM_NAME,
        3 LINE1               CHAR(30),
        3 LINE2             CHAR(30),
      2 AREA                PIC'99',
      2 BUILDING#           PIC'99999',
      2 BUILDING@           PIC'9999',
      2 ROOM_CAPACITY         PIC'999',
      2 #OFASSIGNED         PIC'999',
      2 ROOM_PRIORITY       PIC'9',
      2 BUILDING_PRIORITY   PIC'999',
      2 @OFROOM_DIRECTOR    PIC'9999',
      2 @OFSUPERVISOR(6)    PIC'9999';
```

Figure 5.6 Variable Names of Room File Layout

-Area Code: It defines the area of the examination for the related session.

-Building Number: It is the code which defines the building that the room is located.

-Building Address: It is the address of the building in the Building File that the room is located.

-Room Capacity: It defines the capacity of the room.

-Number of Assigned Candidates: It is the number of candidates confirmed to take the examination.

-Assignment Priority: A priority score is assigned to the rooms depending on their applicability for the examination. Room having higher priority are used first during assignment.

-Address of the Room Director: It is the address of the appointed Room Director in the Staff File.

-Address of the Proctor: It is the address of the appointed Room Proctors in the Staff File.

iv) Building File

It is the file in which the information about all buildings used for examination is stored. The data fields of the records forming this file and their physical names are shown in Figure 5.7 by using Warnier/Orr diagram and Figure 5.8 respectively. The description of data fields is as follows.

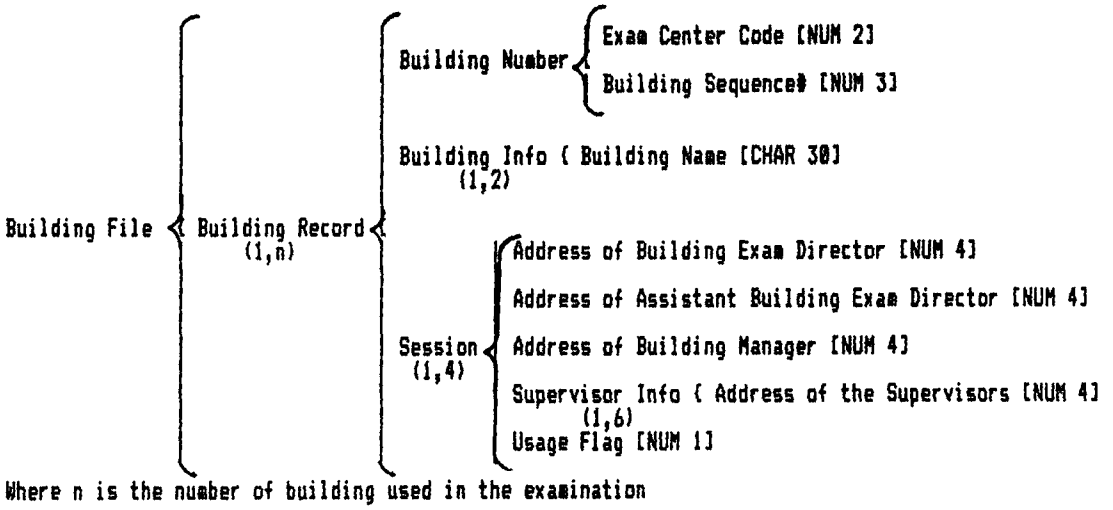


Figure 5.7 Warnier-Orr Diagram of Building File Layout

```
DCL 1 BUILDING_REC,
      2 BUILDING#              PIC'99999',
      2 BUILDING_NAME,
        3 LINE1                CHAR(30),
        3 LINE2                CHAR(30),
      2 SESSION(4),
        3 #OFBUILDING_DIRECTOR PIC'9999',
        3 #OFASST_DIRECTOR     PIC'9999',
        3 #OFBUILDING_MANAGER  PIC'9999',
        3 #OFSTANDBY(6)        PIC'9999',
        3 #OFCANDIDATES        PIC'9999',
        3 #OFROOM_DIRECTOR     PIC'9999',
        3 #OFSUPERVISOR        PIC'9999',
        3 USAGE_FLAG           PIC'9';
```

Figure 5.8 Variable Names of Building File Layout

-Building Number: It is the number which indicates the Examination Center and the order of the building within this center.

-Building Name: Full name of the building.

-Address of the Building Examination Director,

-Address of the Assistant Building Examination Director,

-Address of the Building Manager,

-Address of the Reserve Proctor: These are the addresses of all building staff appointed for all sessions of the examination in the staff File.

-Number of Candidates,

-Number of Room Directors,

-Number of Proctors: These fields are used by the Registration System for staff appointments.

-Usage Flag: It indicates whether the building was used or not in the related session.

v) Optical Files: They are the Magnetic Tape files which are created for transferring the information on the optical forms to REX for Registration, Examination or any other purposes.

vi) Examination Optical File

It is the magnetic tape file on which the answers of the candidates on the answer sheets of the examination are recorded. The data fields of the records forming this file and their physical names are shown in Figure 5.9 by using Warnier/Orr diagram and Figure 5.1Ø respectively. The description of the data fields is as follows.

-Application Number: It is the unique number that defines the candidate. It is used to record the examination results of the candidate to the Examination File.

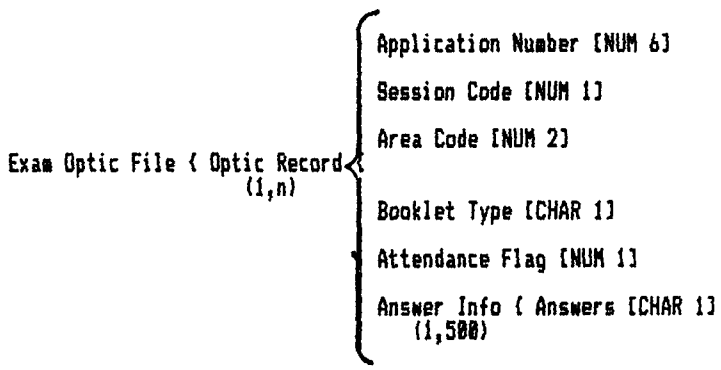


Figure 5.9 W-Orr Diagram of Exam Optical File Layout

```
DCL 1 EXAMOPTIC_REC,
      2 APPLICATION#          PIC'999999',
      2 SESSION               PIC'9',
      2 AREA                  PIC'99',
      2 BOOKLET_TYPE          CHAR(1),
      2 ATTENDANCE            PIC'9',
      2 ANSWERS(500)          CHAR(1);
```

Figure 5.10 Variable Names of Exam Optical File Layout

-Session Code: It is the code which defines the session of the examination that the candidate has taken the examination. If the candidate attends more than one sessions, then the candidate takes more than one answer sheets. Additionally, this code is used to differentiate the answer sheets which might have been assigned more than one for any candidate. This code is a number between 1 and 4.

-Area Code: It is the code which defines the area that the candidate has taken the examination in each session. This code is a number between Ø and 9.

-Booklet Type: It is the type of the booklet. It is generally either "A" or "B".

-Attendance Flag: It is the code which defines whether the candidate has attended the examination or not.

-Answers: They are the answers given by the candidate for the questions of all subareas. The answers can be "A","B","C","D" or "E", " "(space) for no answer, "*"(star) for double answers.

5.2. Layout Files

They are the files designed to keep the physical layouts of the data files and some coded tables which are used in implementation. These files are inserted in the program after file declarations. These layouts are designed for PL/I language. Because of this, they can not be used with any other programming language. A file layout can be prepared for other programming languages and can be added to the system. The names of the coded examination centers, the names of the coded areas used in the examination, the title and duty codes of the staff are used as the table values. These files are kept in the system as <user-system-name>/LAYOUT/<datafilename>.

5.3. Library Programs

Library programs are the procedures to increase the efficiency of the work of the computer system. The programs are written in ALGOL which is the system language. These procedures are compiled and their object codes are loaded to the system. Whenever desired, they can be used with a conformable parameter by using any kind of programming language. The most important characteristic of the library programs is that only one copy of them exists in the memory. So, as they are used by multi users, they are written in a reentrant form.

The library programs exist in the system with the following names.

REX/LIBRARY/<Libname> and
OBJECT/REX/LIBRARY/<Libname>.

The prefix OBJECT, shows that the program consists of executable object code. The programs without this prefix are in source code form. These can be compiled when necessary and their object codes can be reloaded to the system.

The general use of library programs in PL/1 language is as follows.

```
/*----- Library declaration ---------------------*/
DCL <intlibname>
LIBRARY (TITLE=OBJECT/REX/LIBRARY/Libname.')

/*--- Declaration of Procedures in the library --*/
DCL <procedurename>
ENTRY (param-list) OPTIONS (LIBRARY=intlibname);
```

```
/*---- Declaration of Functions in the library --*/
DCL <procedurename>
ENTRY (param-list) RETURNS(value-type)
OPTIONS (LIBRARY=intlibname);

/*----- Usage in the program    -----------------*/
CALL procedurename (param-list) /*procedure call */
XY = procedurename (param-list) /* function call */
```

In implementation, many library programs are used. Some of them are as follows.

-REX/LIBRARY/STARTJOB:This is used to start a job in the system. It is written in ALGOL language because PL/1 does not have this property. It has only one procedure named STARTJOB. This procedure has a parameter whose type is a file name. It starts a job on the system by using the file name, given with the parameter.

-REX/LIBRARY/FILEMANAGER: It is written to execute all duties related with the files, in the system. It has four procedures.

a) COPY: It is written as a procedure. It has two parameters. Both of them are file names. It is used to copy the first file as the second name. If any file given with the second name exists in the system, then old one is removed. If the file given with the first name does not exist in the system, it returns by warning.

b) REMOVE: It is written as a procedure. It has one parameter. This parameter is a file name. It removes the file given with the parameter. If the file to be removed does not exist in the system, it returns by warning.

c) CHANGE: It is written as a procedure. It has two parameters. Both of them are file names. It is used to change the name of the first file as the name of the second one.

d) SEARCH: It is written as a procedure and has two parameters. First parameter is the directory name. The second one is a file name. It finds all files in the directory which are given by the first parameter. It creates the file whose name is given with the second parameter. It records the full names of all existing files to this file. The full names of the files will be "(username) filename ON packname".

-REX/LIBRARY/SUBSYSTEM: It is the library which starts and controls the modules of REX Subsystem. It has one procedure.

a) SUBSYSTEM: It is a procedure which is written to activate one of the three subsystems, forming REX, in the system. It has only one parameter. Depending on this integer parameter, one of the REGISTRATION, EXAMINATION or SUPPORTING subsystems is started.

5.4. Screen Files

They handle the communications with the users during the operation of REX programs. This communication is carried out by forms or menus. In the system, all menus are stored as files. The general format of the names of these files are as REX/SCREENS/n, where n is an integer between 1 and 999.

General structure of screen file layouts is

1 STRUCTn, 2 <screen-data> CHAR (1923);

The screen files are designed for the screen to be used in the system and they contain the control codes usable for these screens. These files are used in order to display data on the screen and to get data from the screen. Besides this, they are used to give different messages by updating on the screen. REX uses several forms/menus. Their names and screen file numbers are shown in Table 5.1.

Table 5.1 (Screen Files of Examination System)

Screen #	Screen Name
Ø	REX_Main Menu
1	REX_Existing User Systems Form
2	REX_User System Parameters Form
4	REX_Subsystems Form
1Ø-199	REX Registration Subsystem Forms/Menus
21Ø	Examination Main Menu
211	Room Assignment Menu
212	Distribution Entry Selection Menu
213	Distribution Entry Form
214	File Creation Menu
215	Evaluation Menu
216	Answer Keys Entry Form 1
217	Answer Keys Entry Form 2
218	Raw Score Evaluation Form
219	Standard Score Evaluation Form
22Ø	Weighted Score Evaluation Form
221	Weigth Entry Form
222	Printout Menu

5.5. Parameter File and Its Usage

These are the files storing detailed information related with the entire Examination Organization, defined by a user system. Their general format of the names in the system are:

<user_system_name>/PARAMETER and
<stationnumber>/PARAMETER.

The file named <user_system_name>/PARAMETER is general parameter file and when the user starts to work on a terminal, its copy with the name <stationnumber>/PARAMETER is prepared.

This file is used to record, the detailed information about the four basic concepts of an examination organization namely the User System, the number of student participating to the examination, the number of active jobs whether the job has been completed normally or not. This file also ensures to obtain general information about the examination organization related to the jobs running under any User system. If the user system is formed for the first time, no parameter file is present and it has to be created at that instance. If this file is present on the system, this implies that a transaction has been made previously on the User System. This file consists of five records differing in appearance from each other. The data fields of the records forming this file and their physical names are shown in Figure 5.11 by using Warnier/Orr diagram and Figure 5.12 respectively.The description of data fields is as follows.

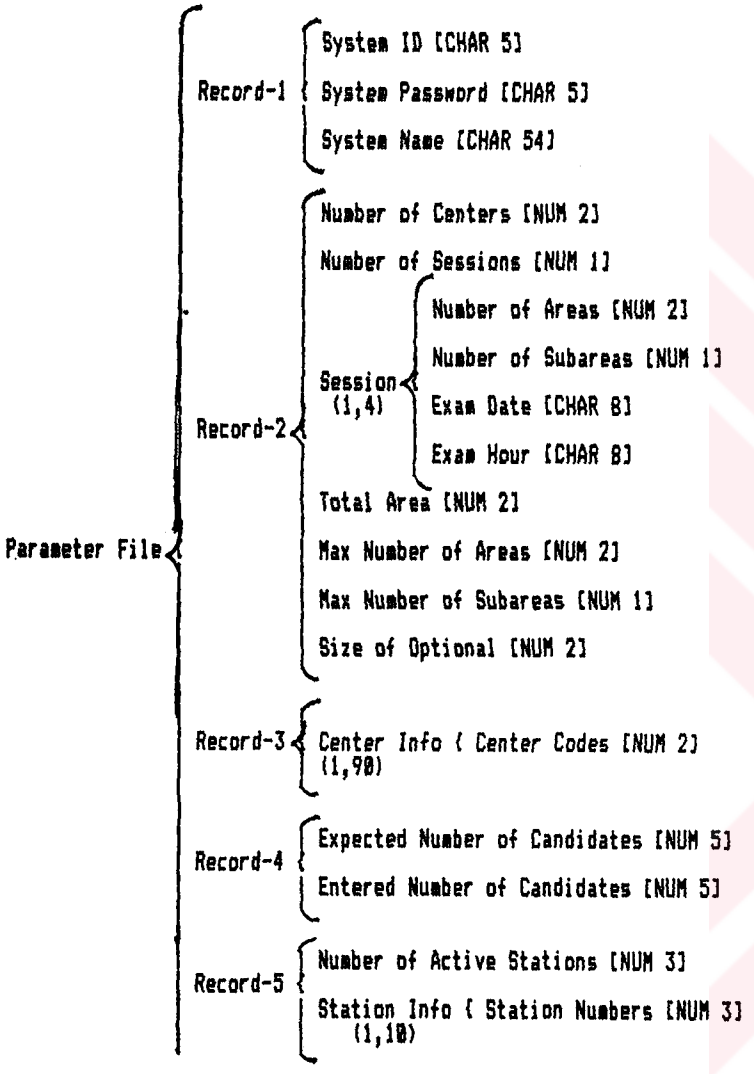


Figure 5.11 Warnier-Orr Diagram of Parameter File Layout

```
1 RECORD_1,
  2 SYSID                 CHAR(5),
  2 SYSPASS               CHAR(5),
  2 SYSNAME               CHAR(54);

1 RECORD_2,
  2 #OFCENTERS            PIC'99',
  2 #OFSESSIONS           PIC'9',
  2 SESSION(4),
    3 #OFAREAS            PIC'99',
    3 #OFSUBAREAS         PIC'9',
    3 EXAMDATE            CHAR(8),
    3 EXAMHOUR            CHAR(8),
  2 TOTALAREA             PIC'99',
  2 MAX#OFAREAS           PIC'99',
  2 MAX#OFSUBAREAS        PIC'9',
  2 SIZEOFOPTIONAL        PIC'99';

1 RECORD_3,
  2 CENTERS(90)             PIC'99';

1 RECORD_4,
  2 EXPECTED              PIC'99999',
  2 ENTERED               PIC'99999',

1 RECORD_5,
  2 #OFSTATIONS
  2 STATIONS(10)            PIC'999';
```

Figure 5.12 Variable Names of Parameter File Layout

Record 1

User System Name: This is a word specifying the user system which has a maximum length of five character. The parameter File name will be "User system Name/PARAMS".

User System Password: This is the password necessary to access the User System structures.

Name in full of the Examination: Name given to the examination.

Record 2

Number of Examination Center: It determines the number of centers in which the examination will be performed.

Number of Session: It determines the number of sessions in which the examination will be performed.

Number of Area: It determines the number of areas in each session.

Number of Subarea: It determines the number of the subareas present in each area of a session.

Examination Date: It determines the date of each session of the examination.

Examination Hour: It determines the hour of each session of the examination.

Record 3

Examination Center Codes: They are the codes given to the centers where the examination will be executed. If the center is a province then provincial traffic codes are used. For the centers other than provinces, the numbers larger than 67 are used.

Record 4

Expected Number of Candidates: The number of candidates expected to be register.

Number of candidates to take the examination: The number of candidates whose admittance for the examination is ensured.

Record 5

Number of Active Stations: The number of jobs on the same User System at a certain time.

Station Numbers: The numbers of terminals which are used in the same user system at a certain time.

5.6. Synchronization Files and Their Usage

During the operation of REX, a synchronization problem comes out when it starts any one of the subsystems. When a module starts operation, the process which starts it, has to wait until the end of module's operation. In the same manner, if a process starts another process, it has to wait for the other's operation. To overcome these problems, the Synchronization Files are used. Their general format of the names in the system is <Stationnumber>/WAITn.

Station number is the working station number and n is the waiting level.

For example: REX is started from the station Ø12. As soon as REX stars the section named REX/EXAMINATION, the Synchronization File with the name 12/WAIT1 is activated. At this stage, if REX/EXAMINATION activates another process, then the synchronization file named 12/WAIT2 will be activated.

These files remains in the system as long as they are required. They are removed when not needed.

5.7. Job Files

They are the files used when the activation of a subsystem is required. Job Files include the commands which provides the running of a program by using a Job Control Language that can be accepted by the computer. These files are executed by Operating System. The library programs, SUBSYSTEM and STARTJOB, are written in order to execute this work via a programming language.

These files are created when they are required, and can be removed when not needed. Their general format of the name in the system is <user-system-name>/ZIPFILE.

5.8. Distribution File

Before the Room Assignment, the distribution of number of students taking the examination in each area of all sessions, which is obtained by the Registration System, is obtained by using the form/menu shown in Figure 5.2Ø. A distribution file is designed in order to use this distribution whenever the work is interrupted. The name of this file in the system is <user_system_name>/DISTRB.

The distribution is recorded on the file by writing the number of students who take the examination in areas of all sessions side by side.

The data fields of the records forming this file and their physical names are shown in Figure 5.13 by using Warnier/Orr diagram and Figure 5.14 respectively.

5.9. Temporary Files.

All candidates declare their areas during registration. As the areas of all sessions are known, the sessions that the candidates will take the examination is already recorded on the Candidate File. By the use of the Candidate File, the candidates are transferred to the Temporary Files depending on the sessions that they will take the examination. If a candidate takes the examination in two sessions, this candidate will exist in the Temporary Files of these two sessions. The names of Temporary Files in the system are as follows.

<user_system_name>/TEMP/1
<user_system_name>/TEMP/2
.
.
.
<user_system_name>/TEMP/<#ofsessions>

By using these files together with the Room Files which are created in accordance with the number of candidates to take the examination in each session, Room Assignment process is performed.

The data fields of the records forming this file and their physical names are shown in Figure 5.15 by using Warnier/Orr diagram and Figure 5.16 respectively.

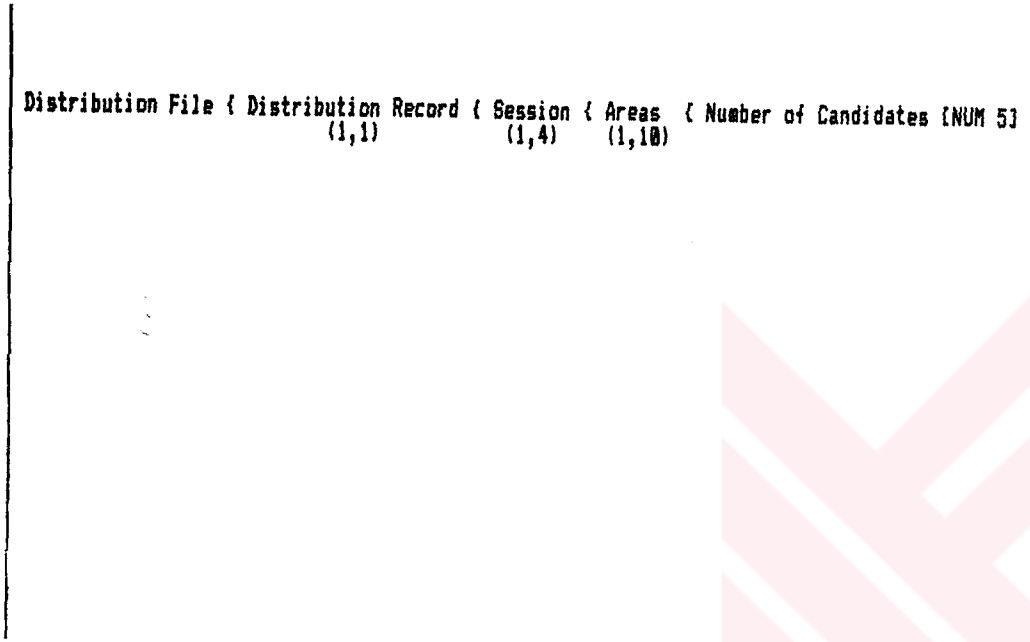


Figure 5.13 W-Orr Diagram of Distribution File Layout

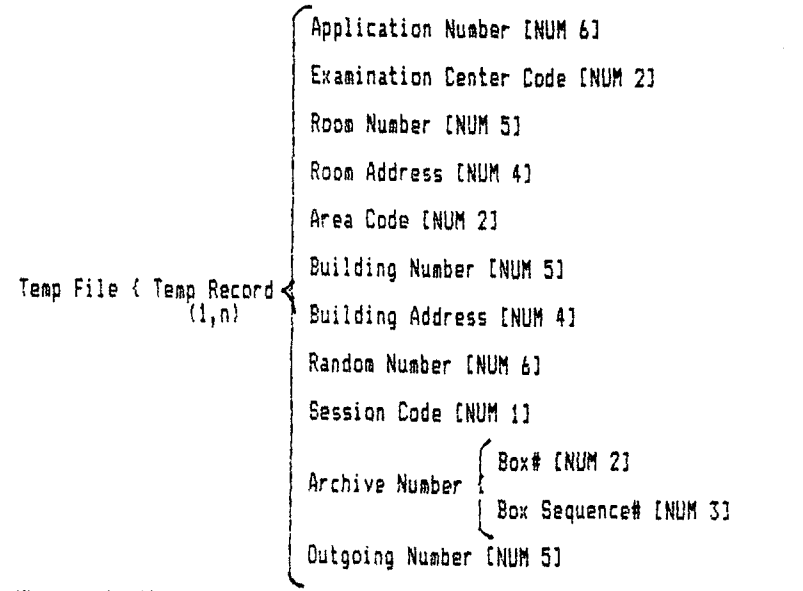


Figure 5.15 Warnier-Orr Diagram of Temporary File Layout

```
DCL 1 DISTRB_REC,
      2 SESSION(4),
        3 AREA(10),
          4 DATA              PIC'99999';
```

Figure 5.14 Variable Names of Distribution File

```
DCL 1 TEMP_REC,
      2 APPLICATION#          PIC'999999',
      2 CENTER                PIC'99',
      2 ROOM#                 PIC'99999',
      2 ROOM@                 PIC'9999',
      2 AREA                  PIC'99',
      2 BUILDING#             PIC'99999',
      2 BUILDING@             PIC'9999',
      2 RND#                  PIC'999999',
      2 SESSION#              PIC'9',
      2 ARCHIVE#,
        3 BOX#                PIC'99',
        3 BOXSEQ#             PIC'999',
      2 OUTGOING#             PIC'99999';
```

Figure 5.16 Variable Names of Temporary File

5.1Ø. Answer Keys File

Before the Examination Evaluation the answer keys of the questions are obtained. These answer keys are recorded on a disk file in order to prevent obtaining these answer keys from the screen again when the user terminates the work. The name of the file in the system is <user_system_name>/KEYS.

Answer keys are recorded on the file in session base. The area code, the position of the subareas in the answer sheets and answer keys for the area in all sessions are recorded on the file.

The data fields of the records forming this file and their physical names are shown in Figure 5.17 by using Warnier/Orr diagram and Figure 5.18 respectively.

5.11. Specification File

During the Examination Evaluation some specifications are provided by the user by means of Forms/Menus. These specifications are stored in a file. The format of the name of this file in the system is <user_system_name>/SPEC. The following information are stored in this file.

-Should the incorrect answers be added during the calculation of the raw score?

-Should the raw score be brought to a certain limit?

-If the raw score will be bring to a certain limit, the coefficient which defines the limit value.

-Should the raw score be converted to standard score?

-Should the weighted score be evaluated?

-If weighted score will be evaluated, the necessary weights

The data fields of the records forming this file and their physical names are shown in Figure 5.19 by using Warnier/Orr diagram and Figure 5.2Ø respectively.

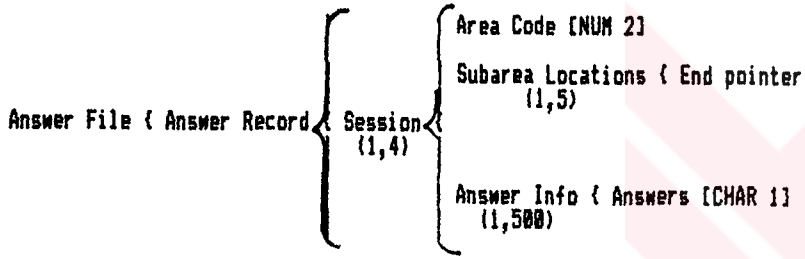


Figure 5.17 W-Orr Diagram of Answer Keys File Layout

```
DCL 1 ANSWER_REC,
      2 SESSION(4),
        3 AREA                PIC'99',
        3 SUBAREA_LOC(5)      PIC'999',
        3 ANSWERS(500)        CHAR(1);
```

Figure 5.18 Variable Names of Answer Keys File

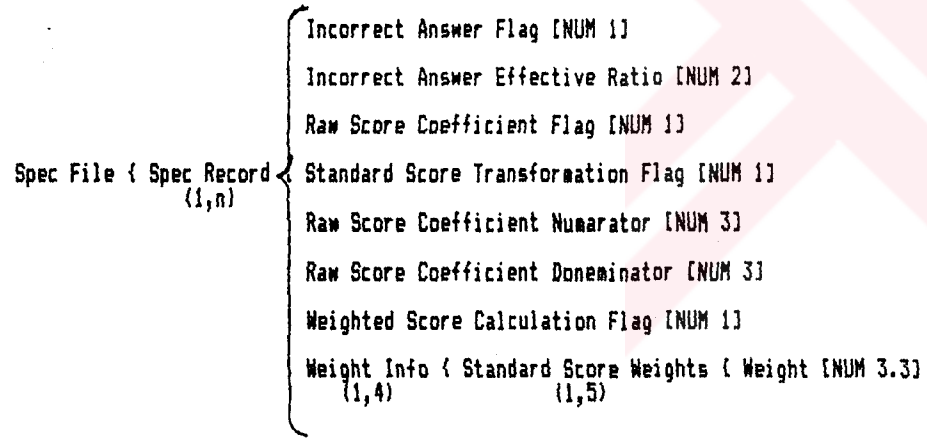


Figure 5.19 W-Orr Diagram of Specification File Layout

```
DCL 1 SPEC_FILE,
      2 IAF                 PIC'9',
      2 IAER                PIC'99',
      2 RSCF                PIC'9',
      2 SSTF                PIC'9',
      2 RSCN                PIC'999',
      2 RSCDN               PIC'999',
      2 WSCF                PIC'9',
      2 SESSION(4),
        3 SUBAREA(5),
          4 WEIGHT          PIC'999.999';
```

Figure 5.2Ø Variable Names of Specification File

6. IMPLEMENTATION OF THE EXAMINATION SYSTEM

The programs which are used in the implementation of the Examination System of which the logical and physical designs are given can be classified as follows according to their functions.

i) The subprograms executing fundamental processes

The programs which are written to realize the Room Assignment,Examination Evaluation and Printing processes that the Examination Subsystem executes. These subprograms utilize many algorithms to execute the processes. Fundamental input/output processes are executed also by these subprograms.

ii) The program sections which handles
the communication between the user and REX

During its operation, REX Examination Subsystem communicates with peripherals, particularly with the user. Before every job, a certain command is entered to REX by the user. REX gives messages to the user about the job or any other subject.

A number of menus are designed for this communication. The program sections which bring these menus to the screen, read and compile when filled by the user and transmit to related sections, are the program sections which provides communication between REX and the user.

iii) Control Programs

REX is a multi-tasking software. This software can also be activated by several users. Multi-tasks can be executed on REX simultaneously. The programs that control the jobs, activate the jobs, provide the control and security of all job files and provide user system level precautions are the control programs.

The user continues his job on one of the Examination, Registration or Supporting subsystems after starting the work with a user system via the Examination Organization System main menu. If REX Examination Subsystem starts operation, the menu shown in Figure 6.1 is displayed on the screen. At this stage, the user may select "Room Assignment", "Examination Evaluation", "Printing" or "Return" options. If the user selects the "Return" option, he returns back to the previous menu. If the user selects one of the above mentioned options other than "Return", it continues from the related section. Three subroutines are designed for these processes.

These subroutines are named as ROOM-ASSIGNMENT, EVAL-OF-EXAM and PRINT-OUT respectively, and they execute all functions in their own structures. These subroutines and their functions are given in detail in the following sections.

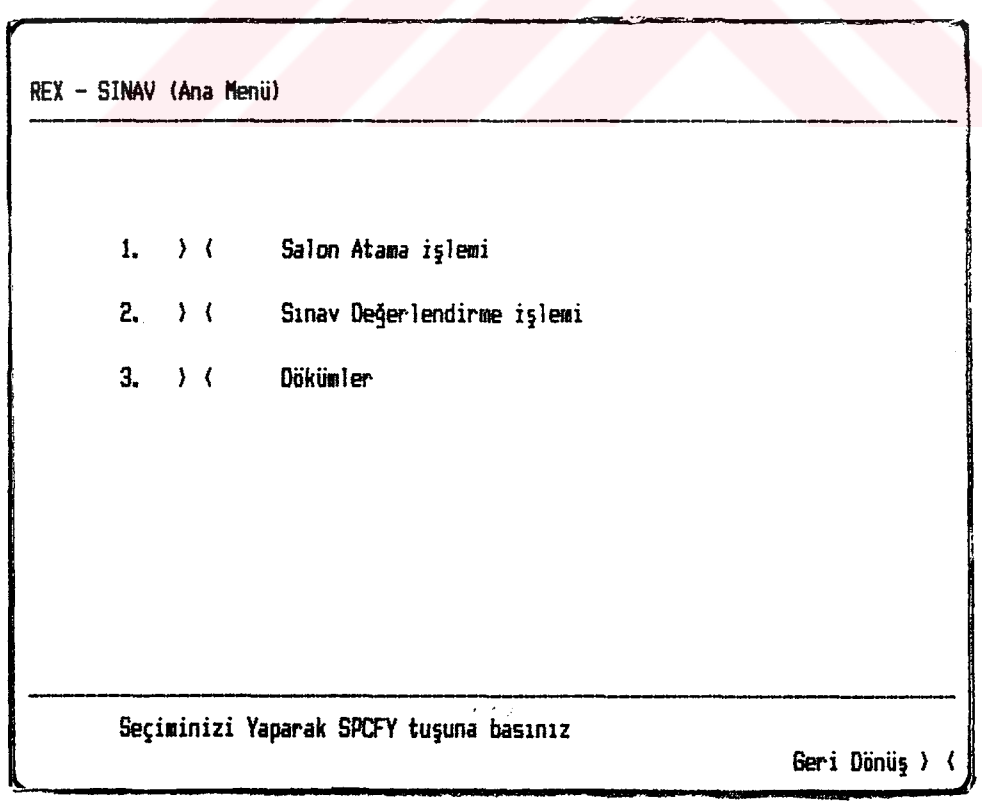


Figure 6.1 Examination Main Menu

6.1. Assignment of the Candidates

The Registration System transfers all information about the examination applicants to the Candidate File, and then it obtains a distribution by using the Parameter File. This distribution includes the number of candidates according to the areas for each session in the examination centers. The user must get this distribution before starting the Room Assignment process. As the variables will always be used in the further sections, it will be useful to explain these variables obtained from the parameter file.

<user_system_name>: The name that specifies the User System.

#OFCENTERS : It determines the number of centers in which the examination will be performed.

#OFSESSIONS : It determines the number of sessions in the examination.

#OFAREAS : It determines the number of areas in each session.

#OFSUBAREAS : It determines the number of subareas in the each session.

CENTERS : It is an array which holds the examination center codes.

In order to implement the room assignment a subroutine is written under the control of the main program. The name of the subroutine is ROOM_ASSIGNMENT. In the ROOM_ASSIGNMENT subroutine, many subroutines are used. The connections between the subroutines are shown in Figure 6.2. All subroutines and their functions are described in following sections. After entering this subroutine a form/menu shown in Figure 6.3 is displayed to the screen. Then user may continue from this menu.

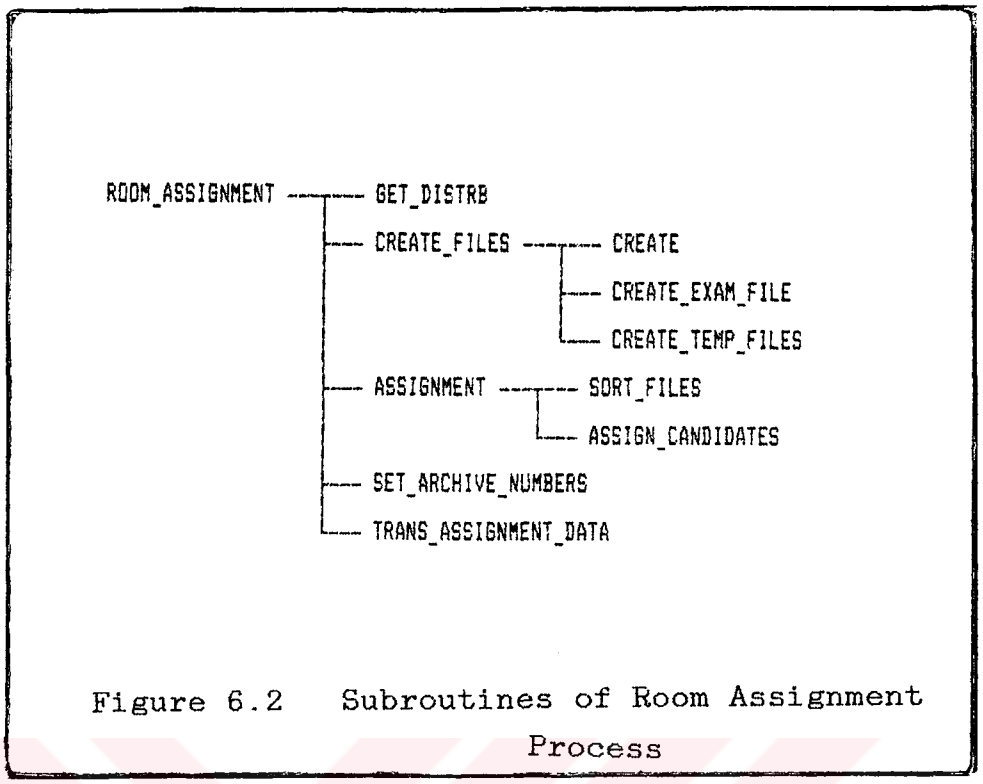


Figure 6.2 Subroutines of Room Assignment Process

REX - SINAV (Salon Atama)

1.) (Aday Dağılımının Girilmesi
2.) (Kütüklerin Yaratılması
3.) (Atamanın Yapılması
4.) (Gidiş Sırano ve Arşiv katsayılarının verilmesi
5.) (Atama Bilgilerinin Aday Kütüğüne Aktarılması

Seçiminizi Yaparak SPCFY tuşuna basınız

Geri Dönüş) (

Figure 6.3 Room Assignment Menu

6.1.1. Getting the Candidate Distribution

As the assignment process starts, a form/menu is displayed for the user to enter the distribution. This form/menu is shown in Figure 6.4. In this form/menu, it is asked to choose one of the options; Should the distribution be entered via the screen or should the preformed file <user_system_name>/DISTRB be utilized? If the file option is chosen, the existance of the <user_system_name>/DISTRB file is checked, if the file exists, this file is used to obtain the distribution data. If the file does not exist, the user is warned and the program continues as if the Screen option is chosen.

If the screen option is chosen, by the use of #OFCENTERS and CENTERS data in the Parameter File, one more form/menu for each center is displayed on the screen. This form/menu is shown in Figure 6.5. By using this form/menu, #OFCENTERS times, a distribution data will be taken for each session.

The numbers taken from the distribution file or screen are kept in the array named ATABLE in the program. The description and meaning of this array in PL/I language is:

```
DECLARE ATABLE (#OFSESSIONS, #OFCENTERS, 1Ø) FIXED;
```

The first dimension of this array is taken as the number of sessions; second one as the number of examination centers and the last one as 1Ø. This is because maximum 1Ø areas can exist in one session. The array holds the number of candidates to take the examination in the related center and the related area of the related session. If the distribution is obtained from screen, a file <user_system_name>/DISTRB is created by using the array named as ATABLE which keeps this distribution. The record layout of this file was given by using Warnier/Orr diagram in Figure 5.13 of section 5.

```
REX - SINAV (Salon Atama-Dağılım)
------------------------------------------------------------------

     1.  ) (    Ekran    (Dağılım ilkdefa girilecek ve kütükte saklanacak)

     2.  ) (    Kütük    (Daha önce girilmiş dağılım kütüğünden alınacak)

------------------------------------------------------------------
     Seçiminizi Yaparak SPCFY tuşuna basınız
                                                   Geri Dönüş ) (
```

Figure 6.4   Distribution Menu

```
REX - SINAV (Salon Atama-Dağılım-Ekran)
------------------------------------------------------------------
 Sınav Merkezi: xx
------------------------------------------------------------------
   1     2     3     4     5     6     7     8     9     10
------------------------------------------------------------------
                           OTURUM 1
 )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   (
------------------------------------------------------------------
                           OTURUM 2
 )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   (
------------------------------------------------------------------
                           OTURUM 3
 )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   (
------------------------------------------------------------------
                           OTURUM 4
 )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   ( )   (
------------------------------------------------------------------
     Dağılım sayılarını girip XMIT tuşuna basınız
     Geri Dönüş için ilgili yere 'G' yazınız          Geri Dönüş ) (
```

Figure 6.5   Distribution Form

6.1.2. Creation of Files

At this stage all processes are carried out with a subroutine called CREATE_FILES. In this subroutine, all files which are to be used for room assignment will be created. When this subroutine is activated, a form/menu shown in Figure 6.6 is displayed on the screen. The subroutine CREATE_FILES which executes this task, uses three subroutines. These three subroutines and their functions are as follows.

6.1.2.1. Creation of Room and Building Files

At this stage all processes are carried out with a subroutine called CREATE_ROOM_BUILD_FILES. After the number of candidates is obtained according to the areas of the sessions, sufficient number of rooms for these candidates must be assigned. General Room File, ATABLE array and general variable values obtained from the Parameter File are used for this purpose.

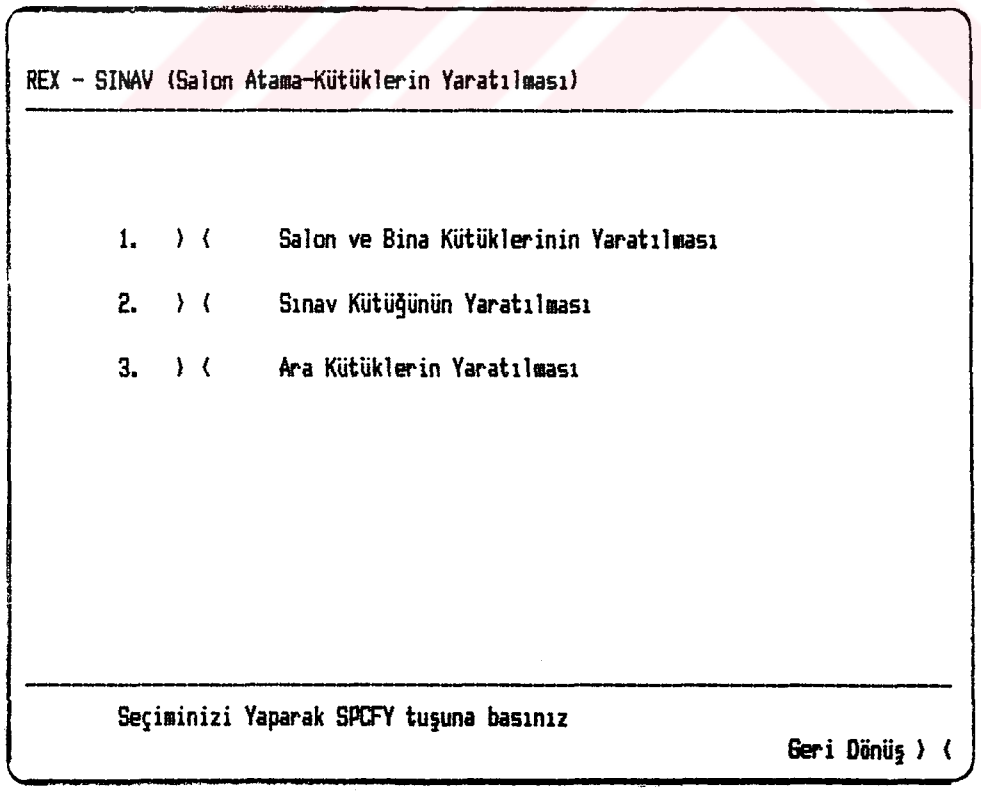


Figure 6.6 File Creation Menu

The rooms to be used for examination must be formed separately for each session. So, there must be same number of Room Files as the #OFSESSIONS. An algorithm which is utilized for creation of Room and Building Files can be fallowed from program listing. According to the algorithm, the total capacity of the rooms will always be more than the number of the candidates. If the equality of the capacity and the number of the candidates is wanted, the necessary condition is defined on the algorithm. However, for the real applications it is better to reserve more capacity than the number of candidates.

During the creation of Room Files, the buildings of the rooms of all sessions are formed in a single file. At the end, a certain number of Room Files which is equal to the #OFSESSIONS and one single Building File will be created.

The format of the names of the created files in the system will be as follows.

<user_system_name>/ROOM/1
<user_system_name>/ROOM/2,
...
<user_system_name>/ROOM/#OFSESSIONS for Room Files.

<user_system_name>/BUILDINGS for Building File.

6.1.2.2. Creation of the Exam File

At this stage all processes are carried out with a subroutine called CREATE_EXAM_FILE. This subroutine creates the Examination File. The name of the created Examination File will be <user_system_name>/EXAM. This file is created by using the Candidate File named as <user_system_name>/CANDIDATE.

The Examination File holds assignment information after Room Assignment, score and answer information after the Examination Evaluation process. After the Examination Evaluation process, the information on the Examination File is transferred to the Candidate File.

During the use of Candidate File, the Examination File of which the physical layout has been given in Figure 5.3 of section 5 is created by including all candidates whose registrations are valid.

6.1.2.3. Creation of the Temporary Files

At this stage all processes are carried out with a subroutine called CREATE_TEMP_FILES. This subroutine creates the Temporary Files which keep the information about the candidates to be used for Room Assignment. During the registration process, the candidates have informed the sessions and areas that they are going to take the examination and this information has been recorded on the Candidate File by the Registration System. The candidates are transferred to the Temporary Files according to the sessions by using the Candidate File. If a candidate takes the examination in two sessions, this candidate will exist in the Temporary Files of these sessions.

The names of the Temporary Files will be as follows:

<user_system_name>/TEMP/1,
<user_system_name>/TEMP/2
...
<user_system_name>/#OFSESSIONS

6.1.3. Assignment Process

At this stage a subroutine called ASSIGNMENT performs the room assignment process by using the Room and Temporary Files which are created by the CREATE_TEMP_FILES and CREATE_ROOM_BUILD_FILES subroutines. Two subroutines are used for room assignment. These subroutines and their functions are as follows.

6.1.3.1. Sorting Files

At this stage all processes are carried out with a subroutine called SORT_FILES. Room and Temporary Files are the files which are used for Room Assignment process. These files need to be sorted before the assignment process. The Temporary Files are sorted according to area code in examination center; and random number in the area code. The purpose of sorting according to the area code is to provide the assignment of the candidates who take the same area in the same room. The random number is used to distribute the candidates in a randomized order. The Room Files are sorted according to the area code in examination center; according to assignment priority in area code and according to the room name in the assignment priority.

The area code is used in sorting for the same purpose used in the Temporary File. The use of assignment priority is to obtain the use of some certain buildings first. Room name is utilized to obtain the use of first floor rooms prior to the other floors.

6.1.3.2. Placement of Candidates to the Rooms

At this stage all processes are carried out with a subroutine called ASSIGN_CANDIDATE. This subroutine assigns the candidates to the rooms by using the Temporary and Room Files.

The room number of the room to which the candidate is assigned and the seat number in the room are recorded to the Examination File named <user_system_name>/EXAM, during assignment. The algorithm which is used in assignment of candidates to the rooms can easily be fallowed from the program listing.

6.1.4. Producing Archive and Outgoing Numbers

At this stage a subroutine called SET_ARCHIVE_NO produces outgoing numbers for answer sheets and some examination documents. It also produces archive numbers to be used for archiving purposes and records these numbers to the Examination File and to the Temporary Files.

6.1.5. Transferring Assignment Data

At this stage all processes are carried out with a subroutine called TRANS_ASSIGNMENT_DATA. This subroutine transfers the candidates' room assignment information from the Examination File to the Candidate File, after the room assignment process.

6.2. Evaluation of the Examination

If the "Examination" option from the REX Subsystem Selection Menu (Figure 3.3) and "Eval_of_Exam" option from the Main Menu of REX Examination Subsystem (Figure 5.1) is chosen, then a new module is activated. The menu which is displayed on the screen at this stage is shown in Figure 6.7.

In this section a subroutine called EVAL_OF_EXAM is used to evaluate the examination of the candidates.

Five subroutines are used in the Examination Evaluation process. The connections between the subroutines are shown in Figure 6.8.

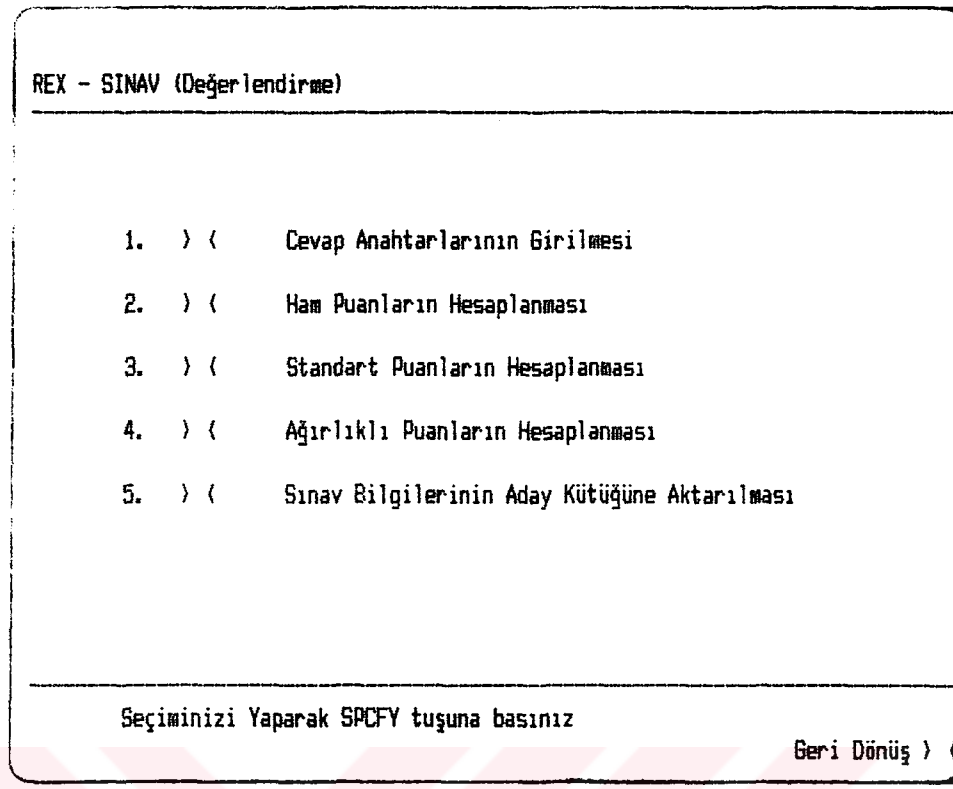


Figure 6.7 Evaluation Menu

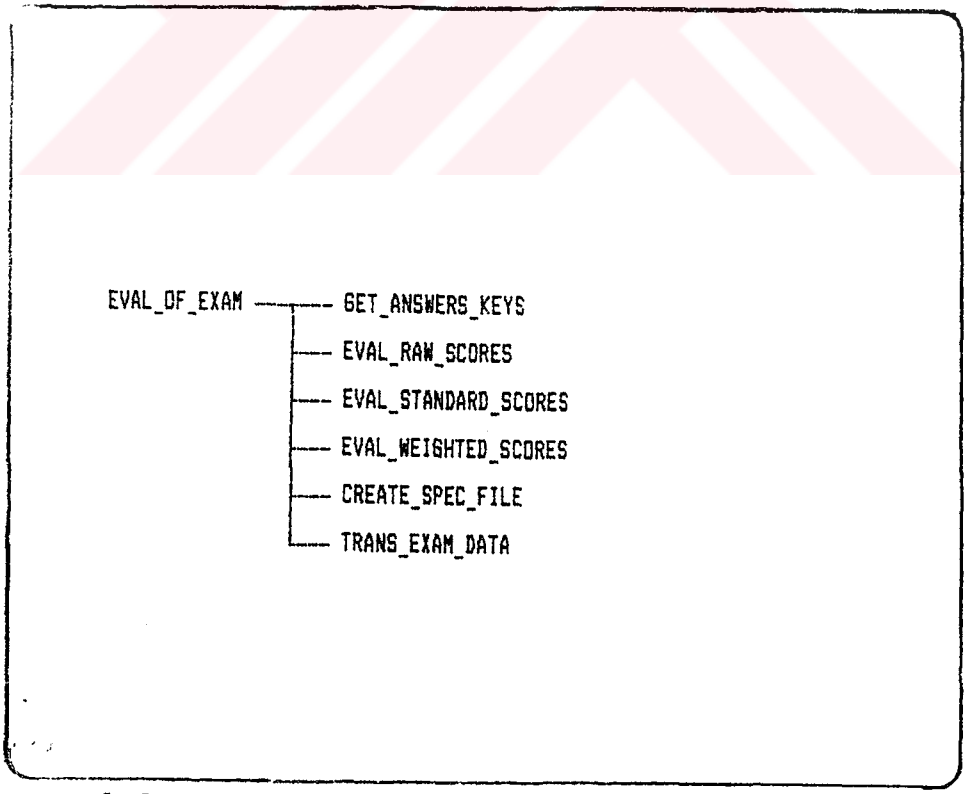


Figure 6.8 Subroutine Structure of Evaluation Process

6.2.1. Getting Answer Keys

Obtaining the answer keys is the first step in the Examination Evaluation. If the evaluation is started for the first time, the data is obtained by two screen form/menus. By the first form/menu shown in Figure 6.9 the locations of subareas on the answer sheets for each session is obtained.

The starting location of the first subarea is 1. The starting location of the other subarea is the next number of the previous subarea's ending location.

This data is kept in KEY_PTRS array. The definition of this array in PL/I language is as follows.

```
DCL
KEY_PTRS    (#OFSESSIONS,MAX#OFAREAS,MAX#OFSUBAREAS)
```

```
REX - SINAV (Değerlendirme-Cevap Anahtarı Girişi 1.menü)

Oturum No : x

Alan Kodu    Altalanların Başlangıç Konumları
             ------------------------------------
    1        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (
    2        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (
    3        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (
    4        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (
    5        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (
    6        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (
    7        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (
    8        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (
    9        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (
   10        1:)  ( 2:)  ( 3:)  ( 4:)  ( 5:)  (

Sayıları girip XMIT tuşuna basınız
Geri Dönüş için ilgili yere 'G' yazınız                Geri Dönüş ) (
```

Figure 6.9 Answers Keys Entry Form 1

The first, second and third dimensions of the array is arranged according to the number of session,the largest number of areas in all sessions, the largest number of subareas in all sessions, respectively. MAX#OFAREAS and MAX#OFSUBAREAS can be found by using the fields named as #OFSESSIONS and array named as #OFAREAS,#OFSUBAREAS in the Parameter File.

After the information obtained by the first form/menu, a second form/ menu shown in Figure 6.1Ø is displayed. By the use of this form/menu, the answer keys of all subareas for each area of all sessions is obtained by having booklet type differences.

This answers keys obtained is kept in A_KEYS array. The definition of this array in PL/I language is as follows.

```
DCL
A_KEYS (BT#,#OFSESSIONS,MAX#OFAREAS,5ØØ) CHAR (1);
```

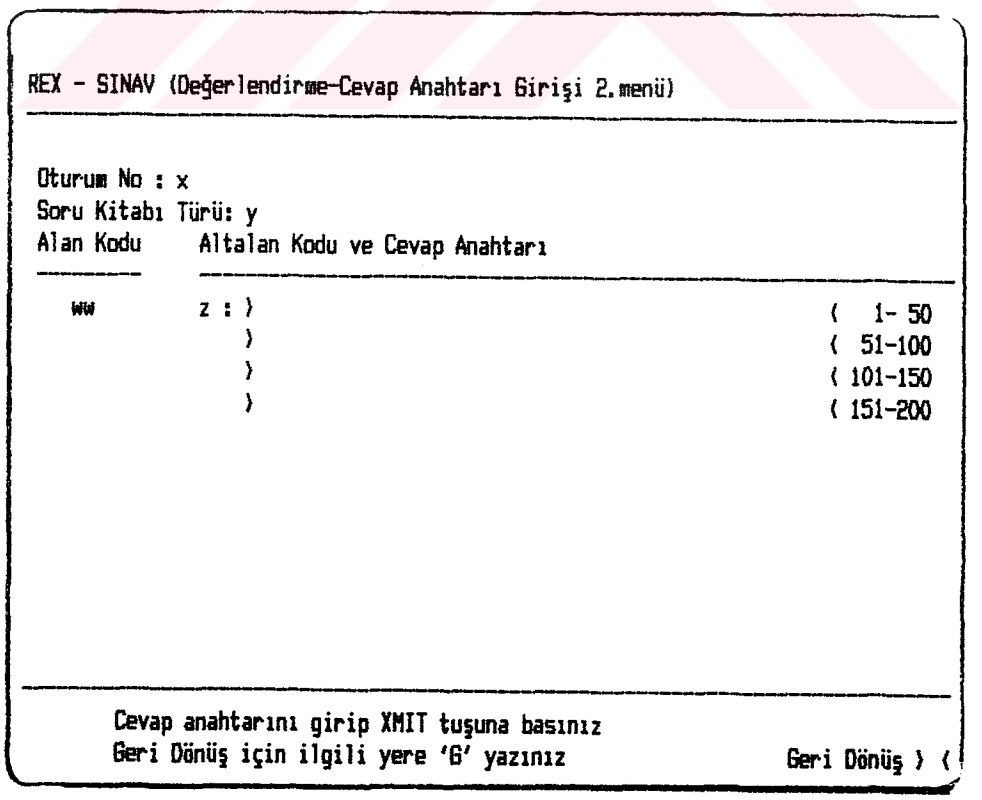


Figure 6.1Ø Answers Keys Entry Form 2

The meaning of dimensions in this array are as follows.

BT#: Number of booklet type.

#OFSESSIONS: Number of sessions in the examination.

MAX#OFAREAS: The largest number of areas in each sessions.

The final dimension is 5ØØ, because there are maximum 5ØØ questions in one area. The location of subareas among these 5ØØ answers exists in the array named as KEY_PTRS. After the answer keys are obtained, a disk file named as <user_system_name>/KEYS is created. The layout of this file by using Warnier/Orr diagram has been given in Figure 5.17 of section 5. The variable names of the file and their meanings are as follows:

AREA_CODE: In the examination organization, a code is given for areas in each session. AREA_CODE is the code which defines this area.

END_PTR: It is the indicator which defines the location of subareas on the answer sheets. It indicates the ending location of every subarea.

ANSWER: It keeps the answer keys for each area. It consists of maximum 5ØØ answers. It keeps the subarea answers together. The answer keys for each subarea is obtained by using END_PTR.

6.2.2. Evaluation of the Raw Scores

After the answer keys obtaining process, the raw scores will be calculated. Before the calculation of raw scores, the following information is obtained from screen or <user_system_name>/SPEC file by the form/menu shown in Figure 6.11.

In the calculation of raw scores the information got from form/menu is as follows.

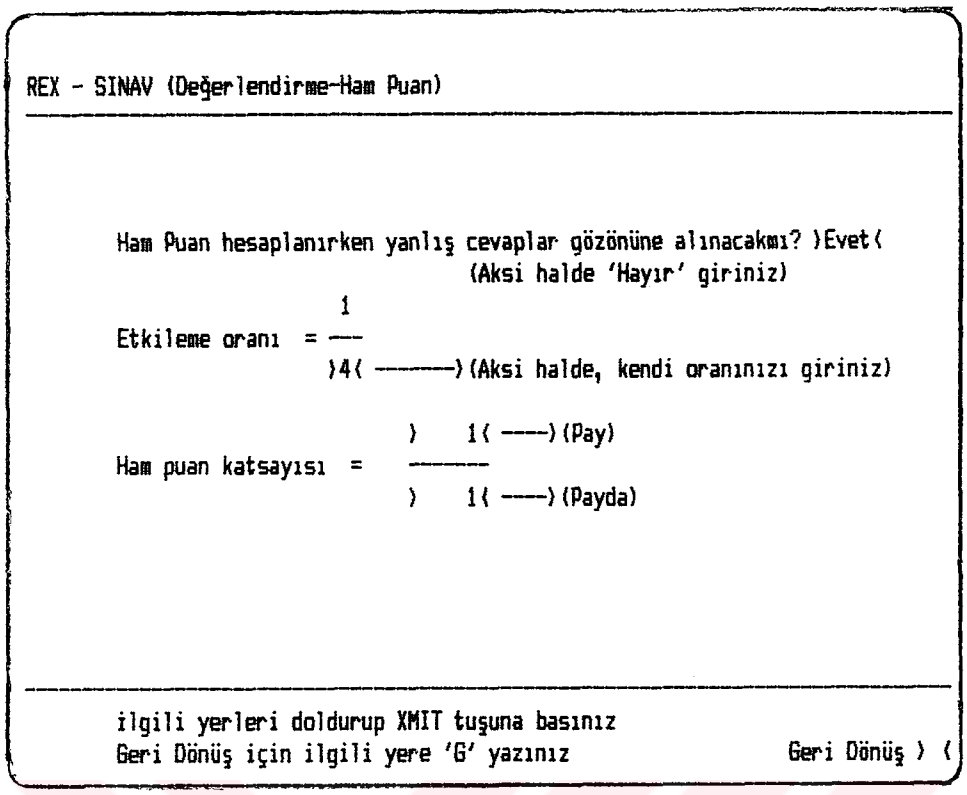


Figure 6.11 Raw Score Evaluation Form

-Whether or not the incorrect answers should be used, if used in what ratio.

-Whether or not a certain coefficient should be used for raw score, if used this coefficient in denominator and numerator form.

The obtained information is stored in order to be used again in case of intermitted operation. For this purpose, the subroutine CREATE_SPEC_FILE is called. When the raw scores are being calculated, the correct and incorrect answers of the candidate for subareas of each area will be counted one by one. By using the number of correct and incorrect answers according to the calculation criteria obtained from the user, a raw score is calculated.

Generally, the formula which is used for calculation of raw score is as follows.

Raw Score=
[#of_correct_answers-k*(#of incorrect answers/n)]*c

k: If the incorrect answers are used= 1
   If the incorrect answers are not used=Ø

n: If k=Ø,then any number different from Ø
   If k=1,than the effective ratio of the number of
          incorrect answers.

c: Is the coefficient used for raw score.
   The aim of using the coefficient is to bring
   the raw scores to certain values.

1/n ratio is the incorrect answers effecting ratio for raw scores, it is generally taken as 1/4.

The raw scores are calculated for subareas of each session's areas. After the calculation of raw scores, an average and standard deviation for each subarea is obtained.

6.2.3. Evaluation of the Standard Scores

After the raw scores are obtained, to determine whether or not the raw scores will be transformed to standard scores, the form/menu in Figure 6.12 or the file <user_system_name>/SPEC is used. If the screen is to be used, the obtained information is recorded on the <user_system_name>/SPEC file. The subroutine called CREATE_SPEC_FILE is used for this purpose.

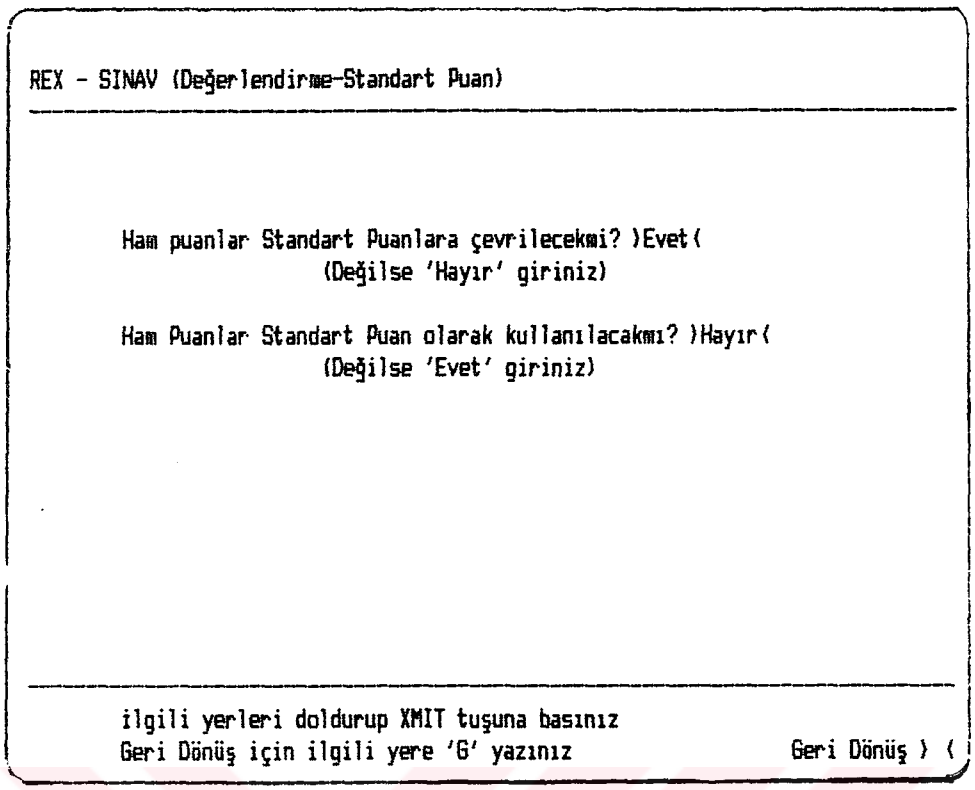


Figure 6.12 Standard Score Evaluation Form

If the raw scores are to be transformed to the standard scores, by using standard deviation, average and raw scores related with the subareas of each area, the standard scores are obtained.

The standard scores are the transformation of the raw scores to standard T scores. Standard T scores have a mean of 5Ø and a standard deviation of 1Ø. The formule used in this transformation is as follows.

standard score =
1Ø * ((raw score-average)/standard deviation) + 5Ø

If the raw scores are not to be transformed to the standard scores, the raw scores are used as standard scores.

6.2.4. Evaluation of the Weighted Scores

In this section a subroutine called EVAL_WEIGHTED_SCORES evaluates a weighted score for each session, after the raw and standard scores for subareas of each session is calculated. However, sometimes this score may be unnecessary depending on the examination type. If the weighted score is to be used, it is required to define the weights of standard scores for subareas of the related sessions. The <user_system_name>/SPEC file or the form/menu shown in Figure 6.13 is used to obtain this information. From this file or form/menu, it is asked whether or not the weighted score will be calculated. If it is desired to be calculated, the weights of the standard scores for subareas of each sessions are obtained by using a form/menu shown in Figure 6.14. If the screen is used, the obtained information is transferred to the <user_system_name>/SPEC file. The subroutine called CREATE_SPEC_FILE is used for this purpose.

The variable called WSC_FLAG defines whether or not the weighted score should be calculated. This is a boolean variable and if its value is "true", it shows that the score is to be calculated.

The weights are stored in an array like the following:

```
DECLARE WEIGHTS (#OFSESSIONS,
                 MAX#OFAREAS,
                 MAX#OFSUBAREAS) FIXED;
```

#OFSESSIONS: Number of sessions

MAX#OFAREAS: The largest number of areas in all sessions.

MAX#OFSUBAREAS: The largest number of subareas in all sessions.

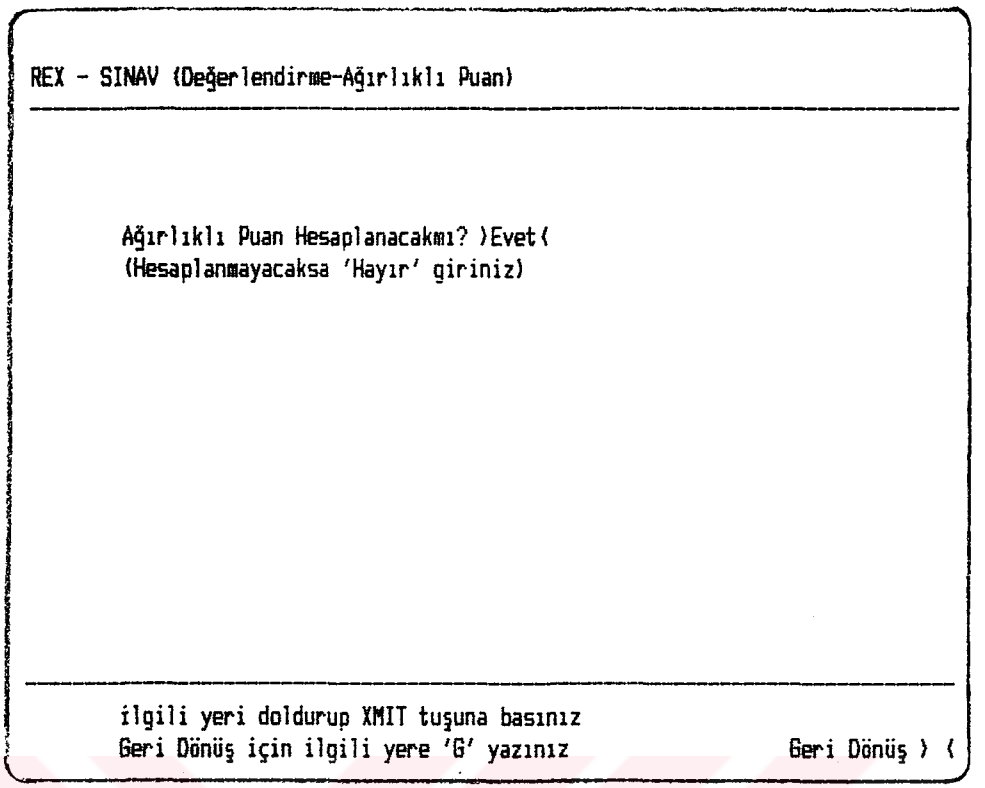


Figure 6.13 Weighted Score Evaluation Form

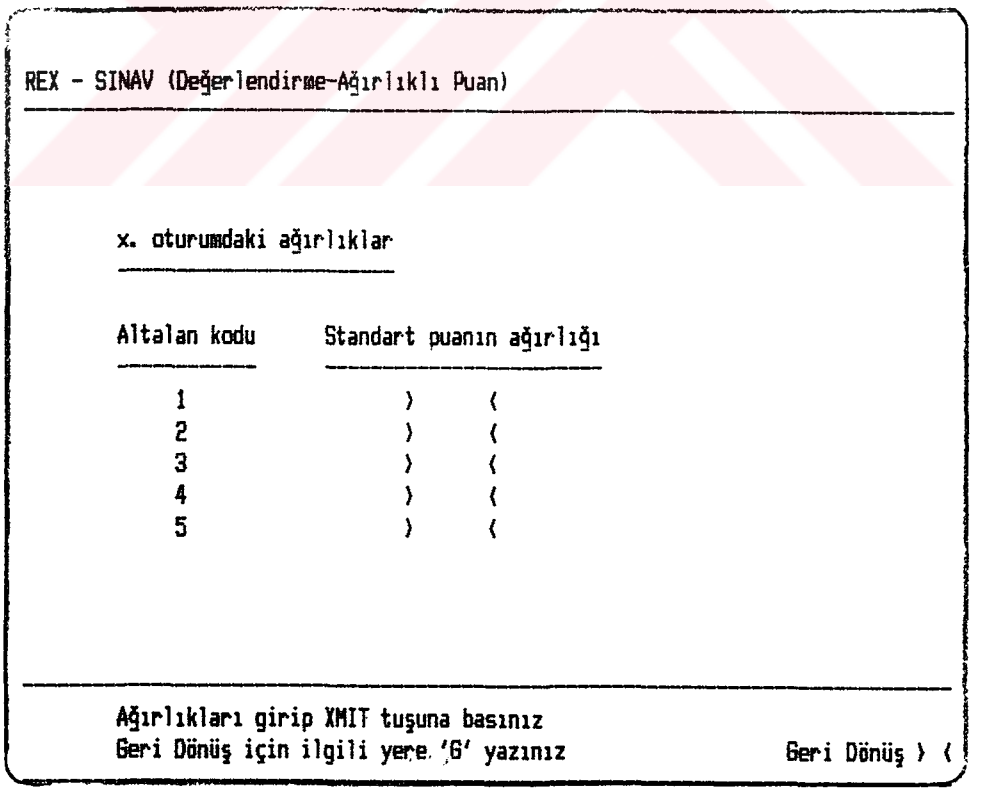


Figure 6.14 Weight Entry Form

If the weighted score is to be calculated, then a weighted score is obtained by using the standard scores in subareas and the weight value in the WEIGHTS array for every session. The formula is as follows:

$$\text{Weighted Score}(i) = \sum_{k=1}^{\#OFSUBAREAS\,(i)} EXAM\_REC.ST\_SCORE(i,k) * WEIGHTS(i, EXAM\_REC.AREACODE(i), k)$$

STANDARD (i,k): It is the standard score of kth subarea in the ith session in the Examination File.

AREACODE (i): It is the area code of the ith session in the Examination File.

WEIGHTS (i,AREACODE(i),k): It is the weight value of the kth subarea of the related area in the ith session in the weight array.

If the weighted score is not to be calculated, then no operation is executed.

6.2.5. Creating Specification File

At this stage all processes are carried out with a subroutine called CREATE_SPEC_FILE. This subroutine is written for serving the subroutines called EVAL_RAW_SCORES, EVAL_STANDARD_SCORES and EVAL_WEIGHTED_SCORED. It records information obtained by these subroutines for evaluation purposes via screen, on a file named as <user_system_name>/SPEC in order to provide for the further use of this information. The layout of this file given with the Warnier/Orr diagram was given in Figure 5.2Ø of section 5. The variable names in the file and their meanings are as follows.

Incorrect Answer Flag(IAF): It determines whether or not the incorrect answers should be taken into consideration in evaluation of raw scores. If its value is zero it means that no information has been processed yet.

If its value is 1,then it means that the incorrect answers will be used, if its value is 2, it means that the incorrect answers will not be used.

Incorrect Answer Effective Ratio(IAER): If IAF=1, it defines the ratio to be used.

Raw Score Coefficient Flag(RSCF): After the calculation of raw score, it defines whether or not it should be multiplicated by a coefficient. If its value is 1, then it will be multiplicated, if its value is 2 then it will not be multiplicated.

Standard Score Transformation Flag(SSTF): It determines whether or not the raw scores should be transformed into standard scores. If its value is Ø, it means no information is processed. If its value is 1, transformation will be performed, if its value is 2, it means no transformation will be performed.

Raw Score Coefficient Numerator(RSCN): If RSCF is 1, it keeps the numerator value of the coefficient.

Raw Score Coefficient Denominator(RSCDN): If RSCF is 1,it keeps the denominator value of the coefficient.

Weighted Score Calculation Flag(WSCF): It determines whether or not the weighted score will be evaluated. If its value is Ø, it means no information has been processed. If its value is 1, it means the score will be calculated, if its value is 2 the score will not be calculated.

WEIGHTS: It keeps the weights of subareas if the weighted score will be evaluated.

Depending on the action of the three subroutines that call this subroutine it reads and sends back data or it records the given data to the file named <user_system_name>/SPEC.

6.2.6. Transferring Evaluation Data

At this stage all processes are carried out with a subroutine called TRANS_EXAM_DATA. This subroutine transfers all information about the evaluation from Examination File to the Candidate File. The transferred information are : for each session, Reading Flag, Examination Attendance Flag, Booklet Type, Weighted Score and Area Code; for each subarea, Number of Correct/Incorrect Answers, Raw and Standard Scores.

6.3. Producing Printouts

At this stage all processes are carried out with a subroutine called PRINT_PRINTOUTS. This subroutine performs all printouts related with Room Assignment and Examination Evaluation processes. It can be activated whenever desired. When it is activated, it displays the form/menu shown in Figure 6.15 on the screen. The user may select any option. Brief description of printouts is as follows.

-Answer Sheet: This sheet is designed as a form that can be read by the optical readers. Application number, name and surname, session code, area code and room information is printed on it by the computer. The candidate codes his/her own number on it. The session code and area code are printed on the form in a way to be read by the optical reader. It is the form on which the candidate will write the answers in the examination.

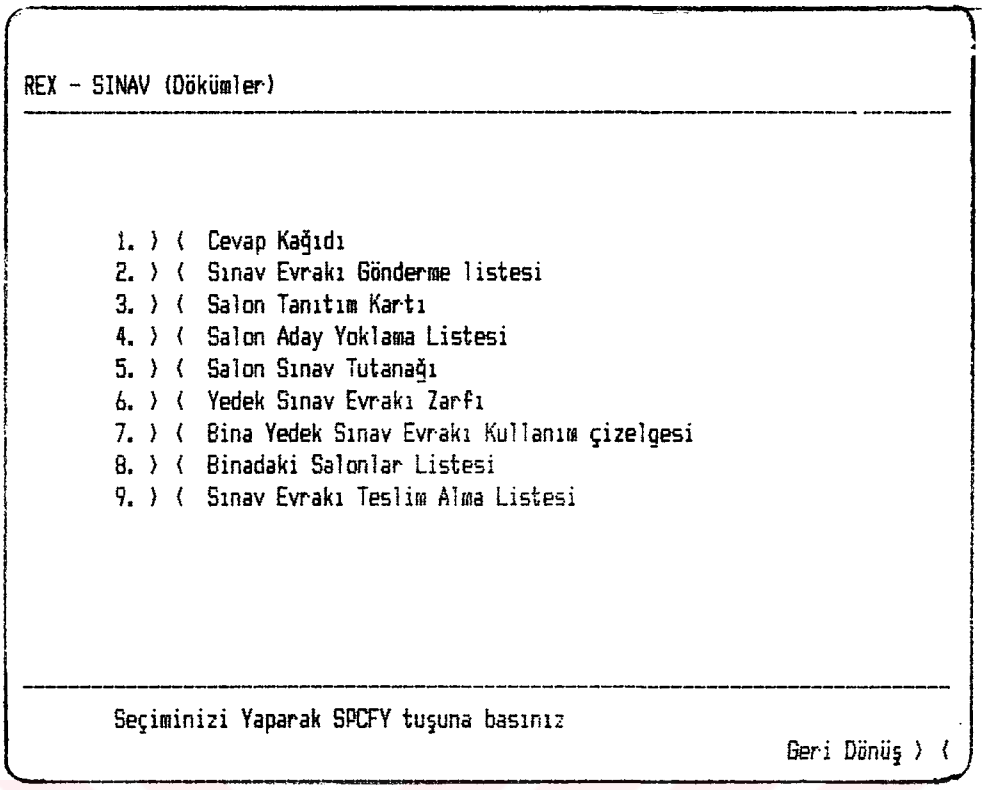


Figure 6.15 Printout Menu

-Room Identification Card: It is the card which identifies the examination room and which is hung on the room doors before the examination. The room number, full name of the room, building number of the room, full name of the building, number of candidates to take the examination, number of staff, full name of the examination and its date are printed on it.

-Room Candidate Attendance List: It is the list on which the candidates to take the examination in the examination room is printed. It is used for attendance checks during the examination. Full name and date of the examination, room number and full name, numbers and names,surnames of the candidates to take the examination, the type of the question booklet used in the examination, and his/her seat number in the examination room are printed on it.

-Room Examination Report: In the examination rooms, it is the document on which the Room Directors write their observations and reports about the examination. Room number and full name, building number and full name, number of answer sheets and question booklets are printed on it.

-Extra Document Envelope: It is the envelope in which the Extra Examination Documents are kept. Building number and full name, name of the Building Examination Director, number of answer sheets and question booklets in it are printed on it.

-Building Extra Documents Utilization Schedule: When the Extra Examination Documents are used, the report about the used examination documents is written on this document. The full name and date of the examination, full name of the examination center, building number and full name of the building are printed on it.

7. TESTING OF REX

REX has been tested on a sample Examination Organization with the following properties.

- User System Name is "ORNEK".
- Number of the Examination Centers is 2.
- Examination Center codes are Ø1 and Ø6.
- Number of Sessions is 2.
- First Session has 4 Areas.
- Second Session has 5 Areas.
- Each Area has 1 Subarea.
- Number of the Candidates is 3Ø8.
- In the first Examination Center,
  - 1 Building, 7 Rooms, 25 Staff
    have been used for the first Session.
  - 1 Building, 9 Rooms, 29 Staff
    have been used for the second Session.
- In the second Examination Center,
  - 1 Building, 6 Rooms, 25 Staff
    have been used for the first Session.
  - 2 Buildings, 8 Rooms, 38 Staff
    have been used for the second Session.

Some of the printouts related to this Examination Organization are given in Appendix F.

8. SUMMARY AND CONCLUSIONS

In this study, a structured information system development for an Examination Organization System is actualized. During planning phase, Examination Organization System is designed to have four subsystems. These subsystems are REX subsystem, Examination Supporting Subsystem, Registration Supporting Subsystem and Test Praperation Subsystem, respectively.

REX Subsystem is the information system section which carries out necessary data processing activities to execute the Examination Organization. Implemented system is the Examination System and divided into two parts. These parts are REX Examination Subsystem and Examination Supportiong Subsystem.

REX Examination Subsystem is the software section of the Examination System and carries out all the necessary data processing activities by using the computer system. Examination Supporting Subsystem is especially designed to provide supporting services for REX Examination Subsystem. Most of these services are dependent on manpower. This subsystem is also an interface between the external environment and the software.

As mentioned before, REX is the software section of the Examination Organization System. REX consists of three subsystems. One of them is REX Examination Subsystem and is handled by this study. Another one is the REX Registration Subsystem and together with the Registration Supporting Subsystem forms the Registration System. Registration System is designed first to obtain the information about the candidates and the executive staff, then to print all the documents to be sent them. This system is handled as a seperate Ms. Thesis. Last subsystem of REX is designed to ease some operations for the user. This subsystem is completely independent from

other two subsystems of REX and does not effect their operations. The name of this subsystem is REX Supporting Subsystem and is not implemented yet.

During the structured information system development study, first the existing system has been analyzed and then, logical design of the system is realized in detail. After logical design, the physical design of the Examination System is implemented.

REX Examination Subsystem is designed to handle three main functions. The first one is the "Room Assignment Process" which selects the sufficient number of rooms for each session according to the candidates in the centers of the examination and places the candidates to these rooms. The second function is the "Examination Evaluation" which evaluates the answers of the candidates and calculates the corresponding scores. The last function is the "Printing Process" which handles the printing of all printouts related to the examination.

The entire software has been loaded and executed on the computer system installed at ÖSYM with sample examination data. The observed results shows that the software is error-free but, since the software was not tested with a real examination data, some unwanted results can be arise during a real application.

Developed software is designed to be modular. Therefore any additions to this software can be done easily. This software is thought to be used in the execution of some Examination Organizations of ÖSYM which are less complex and have a small size. With some additions, the entire software can be used to handle more complex and much bigger applications like ÖSYS (Student Selection and Placement Examination of ÖSYM) project.

REFERENCES

AKTAŞ, A.Z., Structured Analysis and Design of Information Systems, Prentice-Hall, 1987

ATASAGUN, C.A., Design and Implementation of an Automated Registration System, Master's Thesis, 1987

BROOKS, C.H.P. et al., Information Systems Design, Prentice-Hall, 1982

BURCH, J.G., STRATER, F.R., Information Systems: Theory and Practice, John Wiley & Sons, 1984

ÖSYM, İki Aşamalı Üniversitelerarası Öğrenci Seçme ve Yerleştirme Sistemi, 1980

ÖSYM, The System of Student Selection and Placement in Higher Education Institutions in Turkey, 1984

ÖSYM, 1982-1985 ÖSYM Çalışma Raporu, 1985a

ÖSYM, 1985 Öğrenci Seçme ve Yerleştirme Sınavı Başvuru Kılavuzu, 1985b

ÖSYM, 1985 Öğrenci Seçme ve Yerleştirme Sınavı Uygulama Yönergesi, 1985c

ÖZÇELİK, D.A., Test Hazırlama Kılavuzu, ÖSYM, 1981

PAYASLIOĞLU, A., Türkiyede Yükseköğretim Kurumlarına Öğrenci Seçme ve Yerleştirme Sistemi, ÖSYM, 1985

TOKER, F., GÜNALP A., Recent Changes in the System of Higher Education in Turkey, ÖSYM, 1982

WEINBERG, V., Structured Analysis, Prentice-Hall, 1980

YILDIRIM, C., Eğitimde Ölçme ve Değerlendirme, ÖSYM, 1983

YÖK, Yükseköğretimle İlgili Kanun ve Yönetmelikler, 1987

APPENDICES

A. STUDENT SELECTION AND PLACEMENT CENTER (ÖSYM)

REX is designed by the analysis of examination organizations which are applied at ÖSYM and implemented to be used in this center. So, a brief introduction of ÖSYM might be useful.

A.1. ÖSYM Chronicle [ÖSYM, 1984]

ÖSYM, previously called the Interuniversity Student Selection and Placement Center (USYM) was established on 22.11.1974 by the Interuniversity Board in accordance with article 52 of the University Law, No.175Ø. In accordance with the Higher Education Law which came into effect as from 1981, the Center has been attached to the Higher Education Council and its name has been changed to the Student Selection and Placement Center (ÖSYM). According to article 1Ø of the Higher Education Law: "The Student Selection and Placement Center determines, in the context of fundamentals established by the Higher Education Council, the examination principles of the students to be admitted to the institutions of higher education, it prepares the tests, administers them, evaluates them on the basis of their results and the principles determined by the Higher Education Council and in the light of student demands, effects the placement of student candidates in universities and other higher educational institutions, taking into account, as it does so, the students' own preferences, and carries out research related to these activities."

The President of ÖSYM is appointed by the President of the Higher Education Council and is responsible for the administration of the Center. The President presides over the Executive and Advisory Committees. The Executive Committee consists of three members selected for three years by the President of the Higher Education Council among six candidates proposed by the President of ÖSYM. The Advisory Committee is set up as follows: one member

is selected by the Higher Education Council from among two candidates proposed by each university rector and four members are assigned by the Ministry of National Education Youth and Sports. ÖSYM is composed of the following units: The Presidency, The General Secretariat, Information Data Processing, Research, Development and Evaluation, The Examination Services, Library and Documentation, Legal Advisory, and Maintenance. At present a total of 7Ø full-time academic staff work at the Center and there are 35 specialists employed on a part-time basis. Financially ÖSYM is self-supporting. Candidates for places in the higher education institutions pay a registration and an examination fee for the selection and placement examinations. Otherwise, ÖSYM has no other sources of income.

A.2. Organization Schema of ÖSYM

Organization Schema of ÖSYM is shown in Figure A.1. The functions of the units shown in organization schema are as follows [YÖK, 1987].

i) General Secretariat

-To provide the conformability of the center's administrative departments and offices activities concerning the entire center with the related laws, regulations, bylaws and directives.

-To execute the president's correspondences.

-To regulate all protocol, visits and ceremonies of the president.

ii) Data Processing Unit

-To design, regulate and apply the evaluation, classification, selection and placement process, which are performed for higher education student selection and placement.

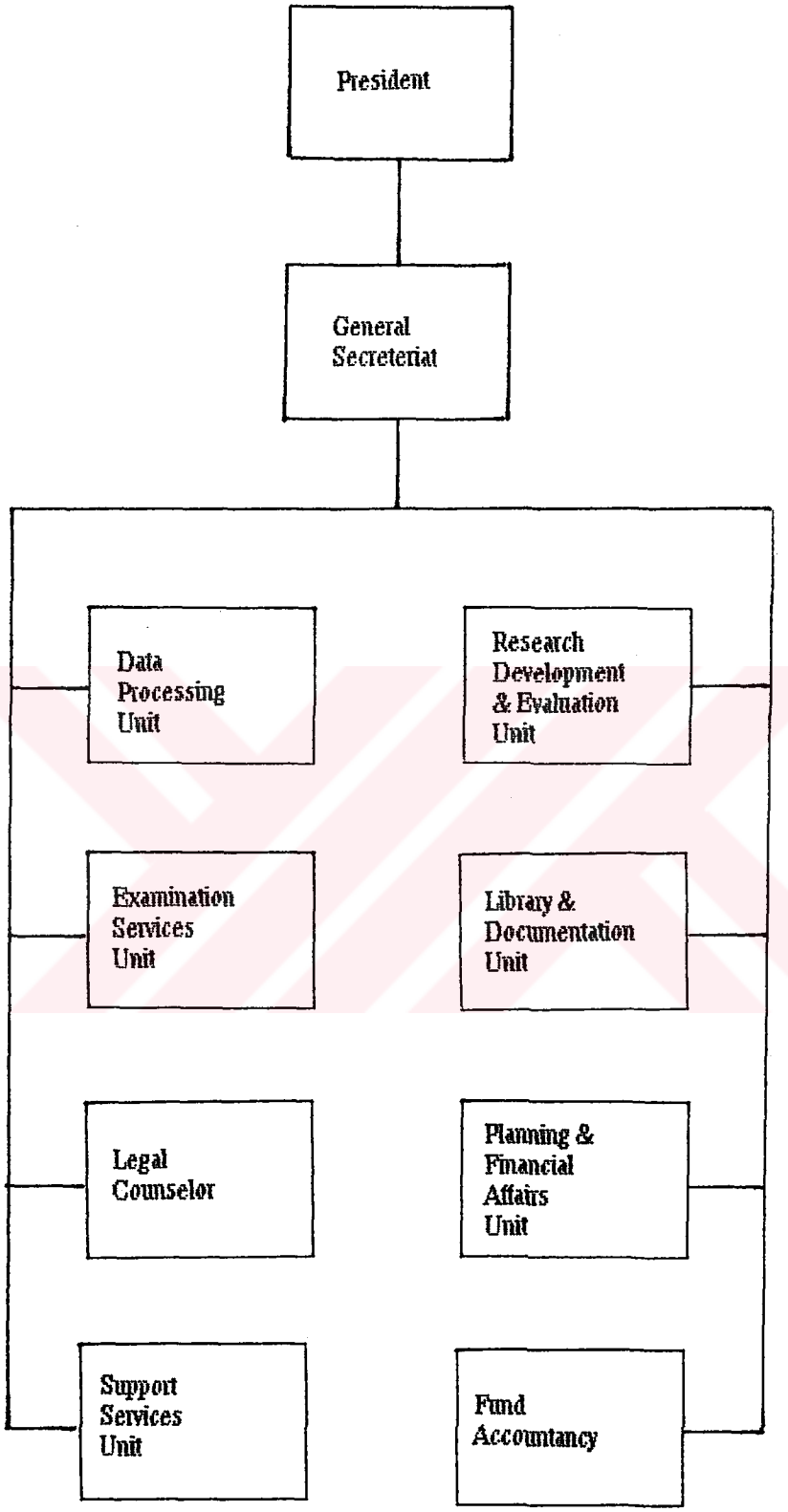


Figure A.1 Organization Schema of ÖSYM

-To provide all necessary data processing services to realize the statistical analysis and researches related with student selection and placement activities.

iii) Research, Development and Evaluation Unit

-To prepare, perform and evaluate the results of research studies to be the basis of higher education institutions' decisions on student selection and placement; or to help or support this kind of research projects.

-To cooperate with the other institutions concerning with the research projects and to realize the education and publication activities related with research results.

-To prepare the tests and similar measuring tools which are to be used for student selection and placement activities and examinations.

-To perform research and development studies in order to increase the validity and reliability of the prepared tests and measurement tools.

-To prepare proficiency tests on various purposes and success tests on intermediate school, high school and undergraduate levels.

-To perform education and publication studies related with the measurement, evaluation and development of these tests.

iv) Examination Services Unit

-To execute application process, examination organization and application services for student selection and placement activities for higher education.

-To perform archives and microfilm duties of documents.

-To handle all kinds of correspondence with the candidates.

v) Library and Documentation Unit

-To execute library activities.

-To collect and protect all kinds of information and documents related with the examinations; and to serve for the users whenever required.

vi) Legal Counselor

-To consult Student Selection and Placement Center in legal matters.

-To take the side of the center in trials.

vii) Planning an Financial Affairs Unit

-To perform the activities on center's administrative and economic subjects.

-To prepare the budgetary forecasts of the center.

-To apply the final budgets.

-To fulfill the requirements of the purchasing regulations.

viii) Support Services Unit

-To execute the related activities of Student Selection and Placement Center personnel's personal rights.

-To perform all technical services of the center, such as maintenance, repairment etc.

-To perform the center's civil defence, security and environment control activities.

-To perform printing and graphics services.

-To carry out lighting, heating, cleaning and similar duties.

ix) Fund Accountancy

-To conduct the Student Selection and Placement Center fund.

-To provide the execution of all activities related with the fund's accountancy.

A.3. Activities of ÖSYM

The services offered by ÖSYM are as follows [ÖSYM, 1984] [PAYASLIOĞLU, A., 1985].

-Student Selection and Placement Examinations for the universities (ÖSYS).

-Foreign Student Examinations (YÖS).

-Registration and examination activities for the Open University (AÖF).

-Central Foreign Proficiency Examinations for Associate Professorship.

-Preparation of special tests for some institutions and evaluation of the results.

-Personnel selection examinations for some public institutions.

-Collection and publication of statistical information on subjects of professional interest.

-Selection examinations for students in the graduate class of high schools in order to give them the facility of education in foreign countries (AFS).

-Selection examinations of the faculty of medicine graduates for specialization studies on medicine (TUS).

-Collection and processing of all statistical data of academic staff and students in the higher education institutions.

B. EXISTING EXAMINATION ORGANIZATION SYSTEM

The activities of an Examination Organization System related with data processing are carried out by the Data Processing Unit of ÖSYM that the organization schema was given Figure A.1. All the documents used in the examination are prepared by this unit. Therefore the functions of Data Processing Unit will be introduced firstly, then the execution of the examination will be presented.

B.1. Data Processing Unit

Functional schema of Data Processing Unit is shown in Figure B.1, overall schema of the Computer System which is installed in the unit is shown in Figure B.2 and overall schema of Optical Reader Systems is shown in Figure B.3. According to the schema B.1 the Data Processing Unit is divided into five subunits. The descriptions of these units are given below.

B.1.1. General Supporting Subunit

The functions of this subunit can be outlined as follows.

-To find solutions for the problems which happens during the use of the computer system by other subunits.

-To take precautions required for more efficient use of the computer system.

-Ensuring the use of any kind of program packages.

-Developing system programs on the Optical Reader.

B.1.2. Candidate Registration Subunit

It is the subunit which performs the activities prior to the examination. The outline of the activities performed by this subunit is as follows.

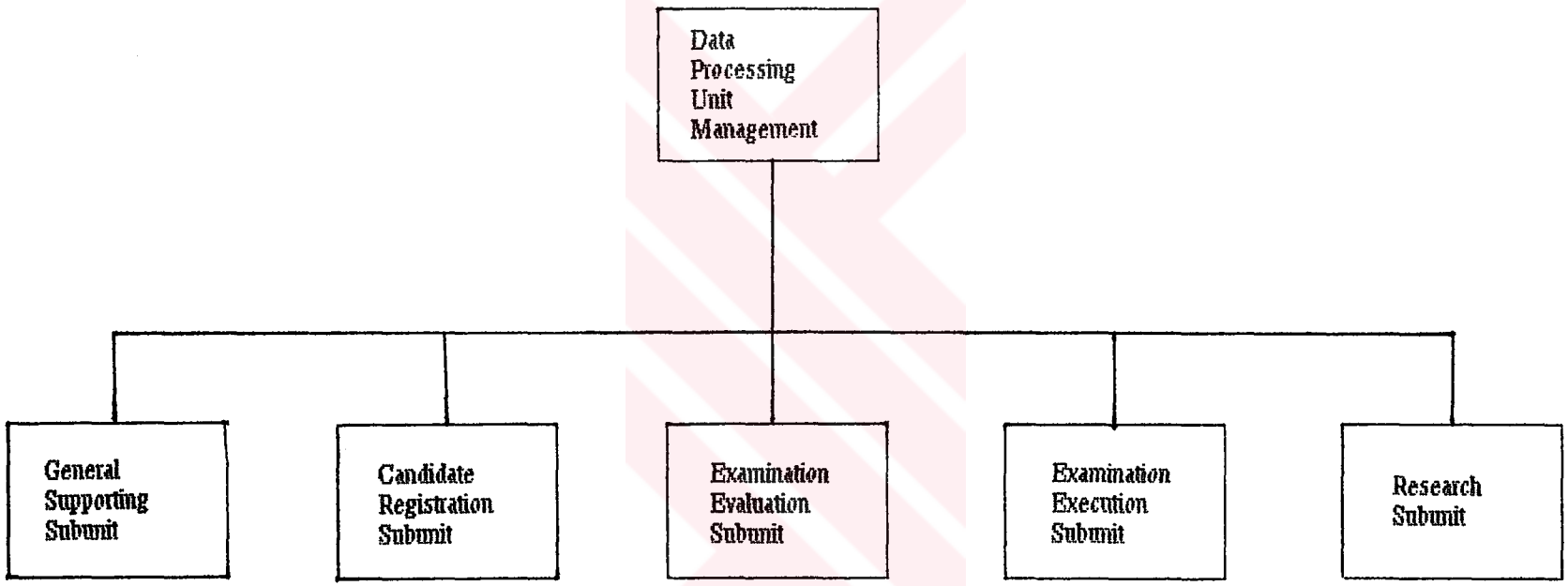


Figure B.1 Functional Schema of Data Processing Unit

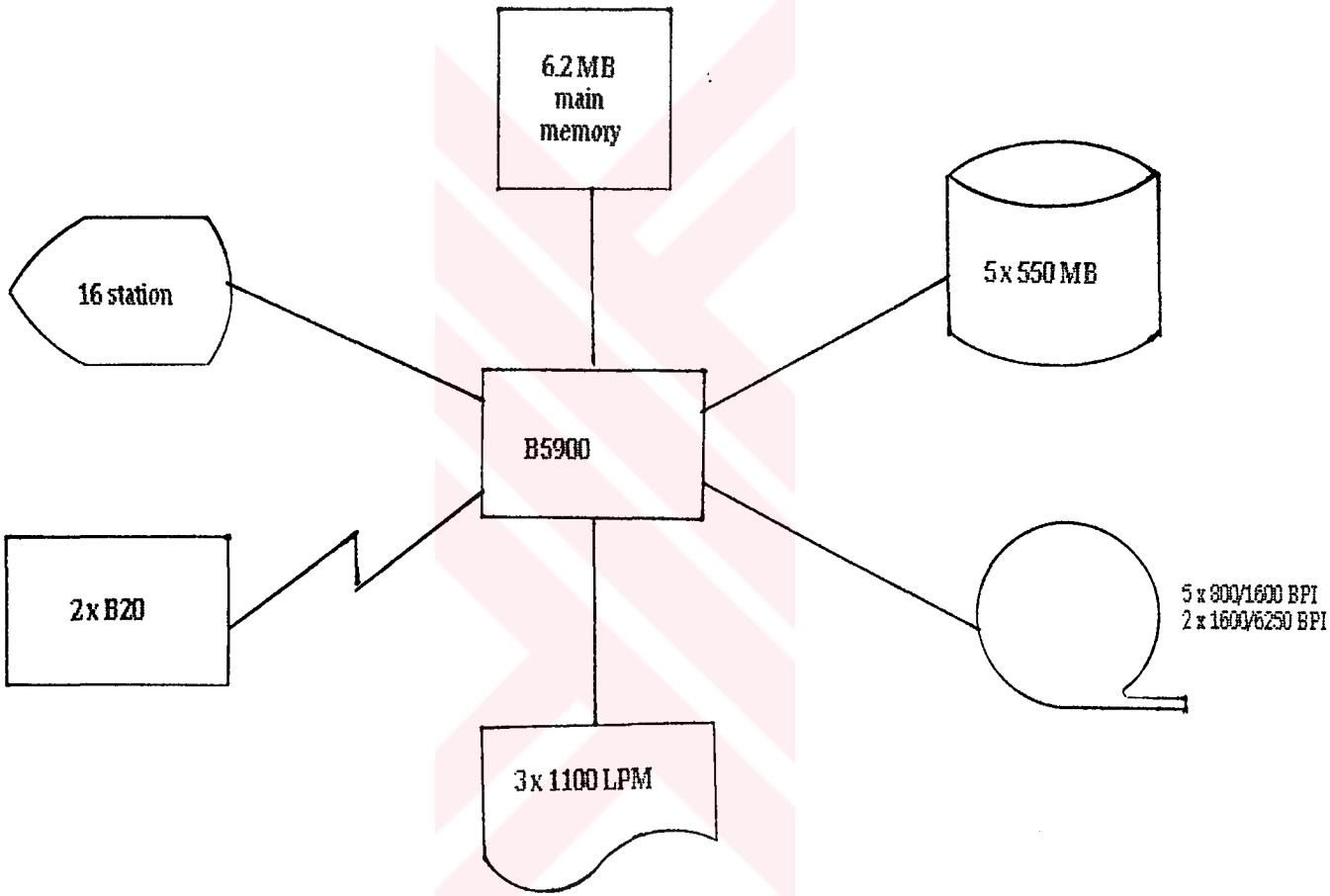


Figure B.2 Overall Schema of Computer System

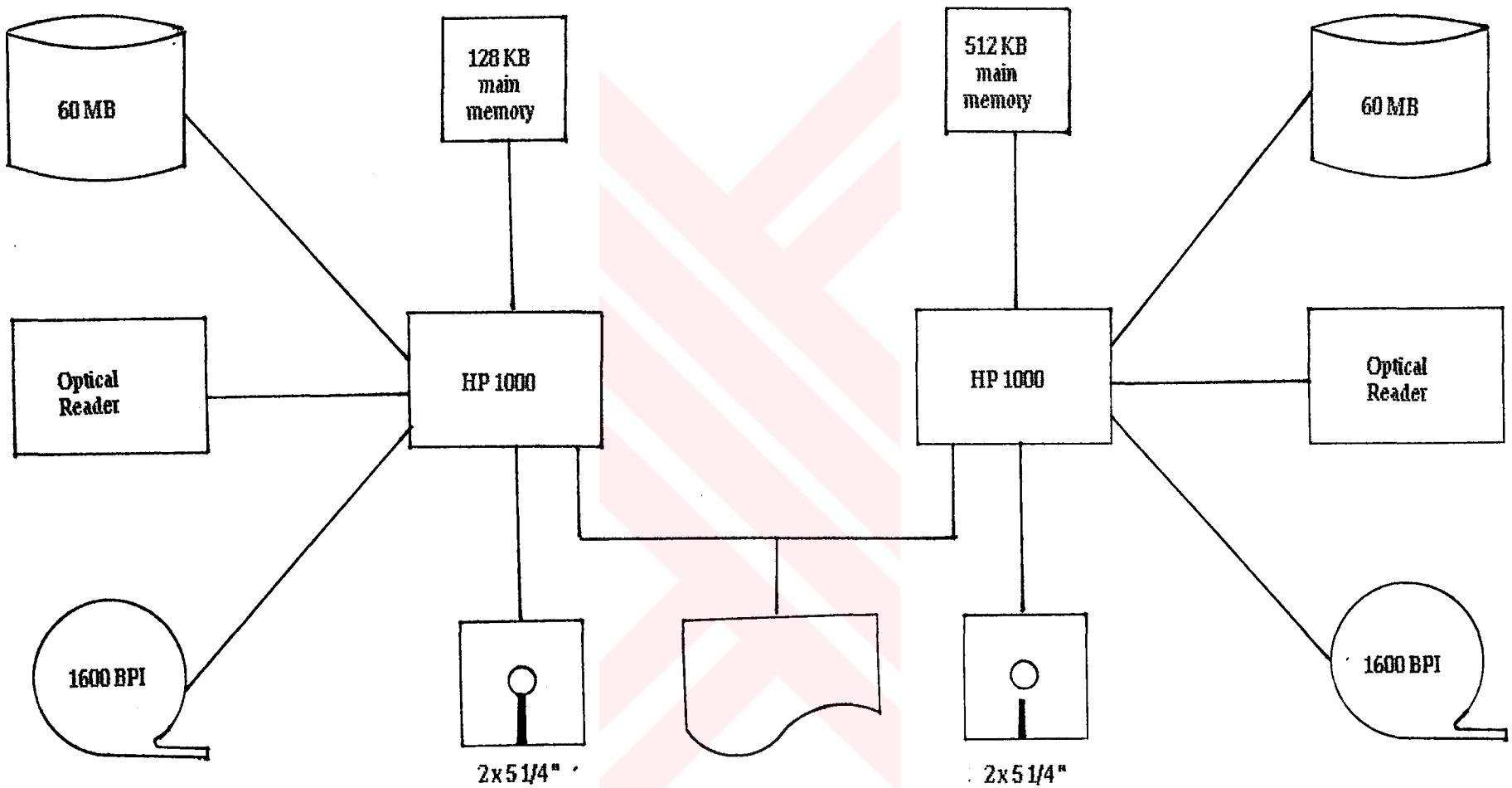


Figure B.3 Overall Schema of Optical Reader Systems

-Design and drawing of the Application Form.

-Preparation of the Application Manual including the basic principles, application conditions concerning the examination and including the information concerning the filling of the Application Form, the documents that will be sent to the candidates, the rules that will be obeyed during the examination, how the examination is evaluated and the means by which the results are announced.

-Printing the application numbers to the Application Forms by means of a computer.

-Ensuring that the Application Manual and Forms has been reached to the candidates.

-Creation of an empty Candidate File.

-Optical reading of the Application Forms incoming from the candidates.

-Transfer of the candidate information obtained with the optical reading to the Candidate File.

-Controlling the information by listing the candidate information recorded in the Candidate File.

-Correction on the Application Forms, and sending back to the optical reading of the documents in which some incomplete or erroneous information has been encountered during the information control process.

-Proceeding to the information controls till the Candidate File will be completely correct.

-Printing of the Examination Entrance Forms (SGB) at the end of the room assignment process realized by the Examination Execution and Evaluation Subunits.

-Printing of the Examination Results Forms (SSB), after that the examination has been realized and the results obtained.

B.1.3. Examination Evaluation Subunit

It is the subunit which performs the room assignment before the examination and evaluation activities after the examination, respectively. The outline of the activities performed by this subunit is as follows.

-Creating the Examination File by means of the Candidate File, in order to assign the candidates to the rooms.

-Supplying the statistical information of the candidates to be assigned in order to ensure the creation of the Room and Building Files by the Execution Subunit.

-Designation of the building and rooms in the examination centers, in which the candidates will take the examination (Room Assignment).

-Transferring the assignment information (Room #, Seat #) at the end of the process to the Candidate File in order to obtain the Examination Entrance Forms by the Candidate Registration Subunit.

-Printing of the Answer Sheets belonging to the candidates that will take the examination, by means of a computer.

-Optical reading of the blank Answer Sheets in order to check whether any kind of fault or dirt are present or not.

-Printing of the Answer Sheet Receipt List in order to ensure the receipt of the Answer Sheets from the examination centers, at the end of the examination.

-Obtaining the answers given by the candidates in the examination, by optical reading of the Answer Sheets received.

-Second optical reading and repairing of the Answer Sheets which have been damaged during the optical reading process.

-Determination of the raw scores, finding the number of the correct and incorrect answers for each test, by confronting the correct answers with those given by the candidates during the examination.

-Calculation of the mean and standard deviation for each test and obtaining the raw score distribution.

-Calculation of the standard scores for each test using the mean and standard deviation.

-Transferring the number of correct and incorrect answers to the Candidate File in order to obtain the raw, standard and weighted scores for printing the Examination Results Form.

B.1.4. Examination Execution Subunit

It executes the activities concerning the staff which performs the examination and the activities belonging to the preparation of the documents used in the examination. The outline of the activities performed by this subunit is as follows.

-Determination of sufficient amount of conformable room and building for each examination center after that the number of the candidates taking the examination has been settled by Examination Evaluation Subunit.

-Sending the lists concerning the information about the characteristics and the number of staff required for the examination buildings and rooms, to the Province Examination Managers after the realization of the room assignment process by the Examination Evaluation Subunit.

-Creating the Staff File after that the staff has been determined by the Province Examination Manager.

-Preparation of the Examination Execution Manual consisting of the rules of the examination and the principles directed to the execution of the examination, in order to be supplied to the examination staff.

-Preparation of the following printouts.

i) Printouts Concerning the Staff

1-Duty Assignment Form
2-Identification Card
3-Official cheque

ii) Printouts concerning the Rooms

1-Room Identification Card
2-Room Candidate Attendance List
3-Room Examination Report

iii) Printouts Concerning the Buildings

1-Building Room List
2-Building Staff List
3-Room Document Receipt/Delivery Report
4-Extra Document Envelope
5-Building Extra Document Utilization Schedule
6-Cheque Distribution Report
7-Cheque Envelope

iv) Printouts Concerning the Province Exam Management

1-Staff and Duty Site List
2-Building Document Receipt/Delivery Report
3-Province Extra Document Utilization Schedule
4-Cheque Envelope Distribution Report

B.1.5. Research Subunit

The outline of the activities performed by this subunit is as follows.

-Evaluation of the inquiries concerning the performed examinations.

-Realization of the subject analysis in order to determine whether the test questions asked to the candidates are appropriate to their aims or not.

-Performing the statistical researches whenever required.

B.2. Execution of the Examination

In the examination organization several documents are used for different purposes in different times. Some of these documents are directed to the candidates and some other to the staff. The execution of the examination is performed by the staff which is liable to perform several tasks. The staff perform their tasks on different site during the execution of the examination [ÖSYM, 1985c].

B.2.1. Staff

The person who has the ultimate authority and responsibility in the examination, is the ÖSYM President. The other authorities in the hierarchy in charge of the execution of the examination are listed below.

i) Province Examination Manager

He has the top level authority and responsibility for the organization and execution of the examination in the province. All the staff related to the examination are appointed by the Province Examination Manager except those assigned by the ÖSYM Presidency.

The Province Examination Manager declares the duties of people whom he appointed by means of a Duty Assignment Form. Besides this, he provides the staff to get the Examination Execution Manual and the Identification Cards on which the name, surname, type and place of work information is written.

ii) Assistant Province Examination Manager

He is the closest companion of the Province Examination Manager. He assists the Province Examination Manager in organization and application of the examination.

iii) Building Examination Director

He is responsible from the preparation of the building, rooms and execution of the examination in all rooms in accordance with the examination terms.

On the examination date, he receives the Building Examination Document from City Examination Courier in the examination building. He gives back the same documents to the City Examination Courier in the examination building after the examination. These documents are as follows.

- Room Document of the Building
- Building Document Receipt/Delivery Report
- Extra Question Booklet Package
- Extra Answer Sheets Package
- Building Staff List
- Examination Execution Manual
- Cheque Envelope

Duties of Building Examination Director can be summarized as follows.

a)He provides the preparation of rooms for the examination by contacting with the Building Manager on time.

b)He provides the placement of Room Identification Cards that he gets from the Province Examination Manager on room doors, at least two days before the examination.

c)He ensures that the Examination Execution Manual is read by all Room Directors in his building.

iv) Assistant Building Examination Director

In the buildings which has more than twenty rooms or more than a thousand candidates, an assistant for the Building Examination Director is appointed. This assistant, cooperates with the Building Examination

Director in all aspects and attends all receiving and returning processes of Building Examination Document.

v) Building Manager

The Building Managers are generally selected from the directors or managers of the institution in the building that the examination will take place. He ensures the preparation of the building and rooms in accordance with examination terms. He helps the Building Examination Director in delivery works of examination documents.

vi) Room Director

The authority and responsibility of executing the examination in accordance with the room rules is given to the Room Director.

His/Her duties can be summarized as follows.

a)To examine the examination room with the Examination Proctors.

b)To assign seat numbers to the seats of the room.

c)To admit the candidates to the examination room by controlling their identification cards.

d)To bring the related examination documents to the room after receiving them from the Building Examination Director. These documents are as follows.

- Answer Sheets Package
- Room Candidate Attendance List
- Room Examination Report
- Question Booklets Package
- Examination Execution Manual

e)To check the attendance by using Room Candidate Attendance List.

f)To provide the distribution of Answer Sheets to the candidates.

g)To read the rules written on the back side of the Examination Execution Manual before starting the examination.

h)To provide the distribution of Question Booklets to the candidates.

i)To inform the candidates about the duration of the examination.

j)To report the necessary points on the Room Examination Report about the candidates who are cheating or the ones who disagree to obey the examination rules during the examination.

k)To provide the collection of Answer Sheets and Question Booklets after the examination.

l)To deliver the Room Examination Document to the Building Examination Director.

vii) Proctor/Reserve Proctor

Under the authority and responsibility of the Room Director, he supervises the execution of the examination in his room in conformance with examination rules. He assists the Room Director in identification card controls, distribution and collection of examination documents to the candidates.

viii) ÖSYM Representative

A representative from ÖSYM is appointed to standardize the examination organization and to help all staff. ÖSYM representatives prepare a report including their observations, evaluations and suggestions and they send these reports to the ÖSYM Presidency.

The ÖSYM Representative is responsible before the ÖSYM President and the Province Examination Manager. If necessary, he can enter each room and control all examination details and candidates' identification cards.

If the ÖSYM representative reports that the examination is not performed in conformance with the examination rules or collective cheating intentions have been occured, then ÖSYM can cancel the results by using its discretionary power.

ix) Province Examination Courier

Appointed by the Province Examination Manager.He receives the examination document bags of his own distribution or collection group's building from the Province Examination Manager on the examination date's morning. He delivers these documents to the Building Examination Director. After the examination, he receives the same documents from the Building Examination Directors and delivers to the Province Examination Management. Delivery process is carried out by signing the Building Document Receipt/Delivery Report which is prepared by ÖSYM and sent to Province Examination Managements.

x) Examination Documents Distribution Courier

He takes the examination documents in closed and sealed trucks from ÖSYM and takes them to the Province Examination Managers. After the examination, he receives the same examination documents from the Province Examination managers and delivers to ÖSYM.

xi) Examination Documents Security Courier

He takes his place in the examination center to which he is appointed on the day before the examination and cooperate with the Province Examination Manager for the security of the examination documents. He assists the Province Examination Manager in collection of examination documents and their delivery to the Examination Documents Distribution Couriers after the examination.

B.2.2. Examination Places

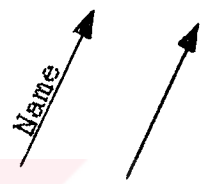
The examination is performed in examination centers. There is one Province Examination Manager for each examination center. The examination is executed in the buildings which are in the center boundaries. There is one Examination Director and Manager for each building. The candidates take the examination in rooms of the building. The Room Director is responsible from the execution of the examination in the room.

C. STRUCTURED TOOLS [AKTAŞ, A.Z., 1987]

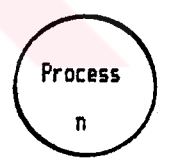
C.1. Data Flow Diagram (DFD)

Data Flow Diagrams can be used either to describe an existing system or a proposed system at the logical level without considering the physical environment. It can be defined as a logical model that describes a system as a network of processes connected to each other and to data stores and also to sources/sinks. The basic symbols that are used in a DFD are as follows.

a) Data Flow
An arrow is used to represent a flow of data. The name of the data flow is written through or next to the line.



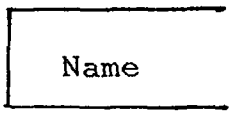
b) Process
A circle represents an automated or manual task or process. It identifies input data that flows into the circle, the transformation that input data undergoes, and the output that flows out of the circle. A brief descriptive statement and a reference number for the process is written inside the circle.



c) Source/Sink
A square is used as an external entity symbol to represent an area where data originates (i.e. source) or terminates (i.e. sink). The name of the entity is written inside the square, in singular. A source produces data flows that our system processes and a sink receives data flows that our system produces.

d) Storage

An open ended rectangle represents a store of information or objects, irrespective of the physical storage medium. The name of the store is written inside the symbol. The store symbol identifies a time delay for its content. If data elements do not flow from one process to the next directly; that is, they are stored for a period of time, this symbol can be used to show such delays.



e) Duplication Symbols

Some sources/sinks or data stores can be drawn more than once on the same DFD in order to minimize the crossing of the data flow lines. The duplicate sources/sinks are identified by a "/" or a "*" symbol. The duplicate data stores are identified by a "|" or a "*" symbol.

f) Additional Symbols

By defining some relational operators, capabilities of a DFD can be improved. These operators are:

* denotes a logical AND connection

⊕ denotes an exclusive OR connection

O denotes an inclusive OR connection

C.2. Warnier/Orr Diagram

Warnier/Orr Diagrams are used to represent data structures as well as processes. The main tool in a Warnier/Orr Diagram is the brace '{' and it shows decomposition of the system. Things that do not decompose further are called elements. If a structure is represented by a Warnier/Orr Diagram, elements are data elements; and if process of a system is expressed, then elements are elementary operations. General form of structures using Warnier/Orr Diagrams are as follows.

a) Hierarchy

The general form of hierarchy is

```
aaa { bb { c
```

which means 'aaa' consists of 'bb' and 'bb' consists of 'c'.

b) Sequence

The general form of sequence is

```
      ⎧ aa
aaa   ⎨ bb
      ⎩ cc
```

which means 'aaa' consists of 'aa' followed by 'bb' and followed by 'cc'.

c) Repetition

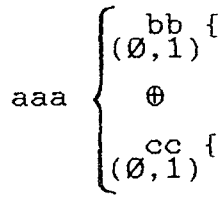
The general form of repetition is

```
aaa { bb
(1,N)
```

which means 'aaa' occurs one to N times.

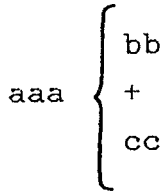
d) Selection

The general form of selection is



e) Concurrency

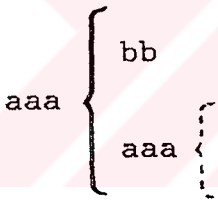
The general form of concurrency is



which means 'aaa' consists of both 'bb' and 'cc' and their order is not important.

f) Recursion

The general form of recursion is



which means 'aaa' consists of 'bb' and itself.

D. GLOSSARY

Başvuru Belgesi	Application Form
Başvuru Kılavuzu	Application Manual
Bina Salon Görevli Gösterim Çizelgesi	Building Staff List
Bina Sınav Evrakı Alındı/Verildi Tutanağı	Building Document Receipt/Delivery Report
Bina Sınav Sorumlusu	Building Examination Director
Bina Sınav Sorumlusu Yardımcısı	Assistant Building Examination Director
Bina Yedek Sınav Evrakı Kullanım Çizelgesi	Building Extra Document Utilization Schedule
Bina Yöneticisi	Building Manager
Binadaki Salonlar Listesi	Building Room List
Cevap Kağıdı	Answer Sheet
Çek Dağıtım Tutanağı	Cheque Distribution Report
Çek Zarfı	Cheque Envelope
Çek Zarfı Dağıtım Tutanağı	Cheque Envelope Distribution Report
Görevli Çeki	Staff Cheque
Görevli Gösterim Çizelgesi	Staff Notification Schedule
Görevlendirme Belgesi	Duty Assignment Form
Görevliler ve Görev Yerleri Listesi	Staff and Duty Sites List
Gözetmen	Proctor
İl Sınav Kuryesi	Province Examination Courier
İl Sınav Yöneticisi	Province Examination Manager
ÖSYM Temsilcisi	ÖSYM Representative
Salon Aday Yoklama Listesi	Room Candidate Attendance List
Salon Başkanı	Room Director
Salon Sınav Evrakı Alındı/Verildi Tutanağı	Room Document Receipt/Delivery Report
Salon Sınav Tutanağı	Room Examination Report
Salon Tanıtım Kartı	Room Identification Card
Sınav Evrakı Dağıtım Kuryesi	Examination Document Distribution Courier
Sınav Evrakı Gönderme Listesi	Examination Document Delivery List
Sınav Evrakı Koruma Kuryesi	Examination Document Security Courier
Sınav Evrakı Teslim Alma Listesi	Examination Document Receipt list
Sınav Sonuç Belgesi	Examination Results Form
Sınav Uygulama Yönergesi	Examination Execution Manual
Sınava Giriş Belgesi	Examination Entrance Form
Yaka Kartı	Identification Card
Yedek Gözetmen	Reserve Proctor
Yedek Sınav Evrakı Zarfı	Extra Document Envelope

APPENDIX E

```
/*                        - - - - - - -                          */
/*                        R E X / S Y S                          */
/*                        - - - - - - -                          */
SYSTEM:PROC OPTIONS(MAIN);

DCL R FILE RECORD ENV(KIND='DC',MAXRECSIZE=1923,FRAMESIZE=8,
                      BLOCKSIZE=1923,MYUSE='IO');
DCL
( NUL,SOH,STX,ETX,EOT,ENQ,ACK,BEL,BS,HT,LF,VT,CLS/* FF */,
  CR,REVERSE/* SO */,UNDERLINE/* SI */,DLE,DC1,DC2,DC3,DC4,
  NAK,SYN,NORMAL/* ETB */,BLINK/* CAN */,SECURE/* EM */,
  BRIGHT/* SUB */,ESC,FS/* WRITE ONLY */,GS/* RIGHT JUST */,
  RS/* END FIELD */,US/* LEFT JUST */ ) CHAR(1);

UNSPEC(      NUL)='00000000'; UNSPEC(      SOH)='00000001';
UNSPEC(      STX)='00000010'; UNSPEC(      ETX)='00000011';
UNSPEC(      EOT)='00111110'; UNSPEC(      ENQ)='00101101';
UNSPEC(      ACK)='00101110'; UNSPEC(      BEL)='00101111';
UNSPEC(       BS)='00010110'; UNSPEC(       HT)='00000101';
UNSPEC(       LF)='00100101'; UNSPEC(       VT)='00001011';
UNSPEC(      CLS)='00001100'; UNSPEC(       CR)='00001101';
UNSPEC(  REVERSE)='00001110'; UNSPEC(UNDERLINE)='00001111';
UNSPEC(      DLE)='00010000'; UNSPEC(      DC1)='00010001';
UNSPEC(      DC2)='00010010'; UNSPEC(      DC3)='00010011';
UNSPEC(      DC4)='00111100'; UNSPEC(      NAK)='00111101';
UNSPEC(      SYN)='00110010'; UNSPEC(   NORMAL)='00100110';
UNSPEC(    BLINK)='00011000'; UNSPEC(   SECURE)='00011001';
UNSPEC(   BRIGHT)='00111111'; UNSPEC(      ESC)='00100111';
UNSPEC(       FS)='00011100'; UNSPEC(       GS)='00011101';
UNSPEC(       RS)='00011110'; UNSPEC(       US)='00011111';

DCL DC2ETX CHAR(2);   DC2ETX=DC2!!ETX;

DCL ASCII CHAR(80)
    INIT(' !"#$%&''()*+,-./0123456789:;<=>?@ABCDEFG'!!
         'HIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmno'),
    POS(80) CHAR(1) DEF ASCII POS(1);

% INCLUDE 'REX/SCREENS/0.';
% INCLUDE 'REX/SCREENS/1.';
% INCLUDE 'REX/SCREENS/2.';
% INCLUDE 'REX/SCREENS/4.';

DCL U FILE RECORD ENV(KIND='DISK',MAXRECSIZE=180,AREAS=5,
                      AREASIZE=1);

DCL 1 RECORD0,
      2 SYSID            CHAR(5),
      2 SYSPASS          CHAR(5),
      2 SYSNAME          CHAR(54);
DCL 1 RECORD1,
      2 #OFCENTERS       PIC'99',
      2 #OFSESSIONS      PIC'9',
      2 SESSIONS(4),
        3 #OFAREAS       PIC'99',
        3 #OFSUBAREAS    PIC'9',
        3 EXAMDATE       CHAR(8),
```

```
        3 EXAMHOUR        CHAR(5),
      2 TOTALAREA         PIC'99',
      2 MAX#OFAREAS       PIC'99',
      2 MAX#OFSUBAREAS    PIC'9',
      2 SIZEOFOPTIONAL    PIC'999';
DCL 1 RECORD2,
      2 CENTERS(90)       PIC'99';
DCL 1 RECORD3,
      2 EXPECTED          PIC'99999',
      2 ENTERED           PIC'99999';
DCL 1 RECORD4,
      2 #OFSTATIONS       PIC'99',
      2 STATIONS(10)      PIC'9999';

ON RECORD(U);

GETUSERPARAMS:PROC(SYSID);
   DCL SYSID CHAR(*);
   TITLE(U)=SYSID !! '/PARAMETERS';
   OPEN FILE(U) INPUT;
   READ FILE(U) INTO(RECORD0);
   READ FILE(U) INTO(RECORD1);
   READ FILE(U) INTO(RECORD2);
   READ FILE(U) INTO(RECORD3);
   READ FILE(U) INTO(RECORD4);
   CLOSE FILE(U) ENV(LOCK);
END GETUSERPARAMS;

DCL LIB1 LIBRARY(TITLE='OBJECT/REX/LIBRARY/SUBSYSTEM.'),
    LIB2 LIBRARY(TITLE='OBJECT/REX/LIBRARY/FILEMANAGER.');

DCL SUBSYSTEM ENTRY(FIXED) RETURNS(BIT(1))
              OPTIONS(LIBRARY=LIB1),
  ( COPY      ENTRY(CHAR(*),CHAR(*)) RETURNS(BIT(1)),
    CHANGE    ENTRY(CHAR(*),CHAR(*)) RETURNS(BIT(1)),
    SEARCH    ENTRY(CHAR(*),CHAR(*)) RETURNS(BIT(1)),
    REMOVE    ENTRY(CHAR(*)) RETURNS(BIT(1)) )
              OPTIONS(LIBRARY=LIB2);

DCL 1 STRUCT0,
      2 SYSID             CHAR(5),
      2 SYSPASS           CHAR(5);
DCL 1 STRUCT2,
      2 SYSNAME           CHAR(54),
      2 #OFCENTERS        PIC'99',
      2 CENTERS(90)       PIC'99',
      2 #OFSESSIONS       PIC'9',
      2 SESSIONS(4),
        3 #OFAREAS        PIC'99',
        3 #OFSUBAREAS     PIC'9',
        3 EXAMDATE        CHAR(8),
        3 EXAMHOUR        CHAR(5),
      2 EXPECTED          PIC'99999',
      2 CONTROL           CHAR(1);
DCL 1 STRUCT4,
      2 SPCFY(4)          CHAR(1);
```

```
DCL MESSAGE                CHAR(200) VAR,
    MYSTA                  PIC'9999',
  ( CONTINUEFLAG,
    NOTSPCFYFLAG,
    RETURNFLAG )           BIT(1),
    MYSELF                 BUILTIN;

ON RECORD(R);

MYSTA=STATION(MYSELF) * -1;

DO WHILE(1);
   CONTINUEFLAG='1'B;
   DO WHILE(CONTINUEFLAG);
      WRITE FILE(R) FROM(SCREEN0);
      READ FILE(R) INTO(STRUCT0);
      IF STRUCT0.SYSID = 'QUIT' THEN DO;
         CALL REMOVE(MYSTA !! '/PARAMETERS');
         CALL DELSTATION(MYSTA);
         STOP;
      END;
      IF SUBSTR(STRUCT0.SYSID,1,1) = ESC THEN DO;
         CALL FINDUSERSYS;
         WRITE FILE(R) FROM(SCREEN1);
         READ FILE(R) INTO(MESSAGE);
      END; ELSE
         CONTINUEFLAG='0'B;
   END;
   RETURNFLAG='0'B;
   TITLE(U)=EXCEPT(' ',STRUCT0.SYSID) !! '/PARAMETERS';
   IF ¬ RESIDENT(U) THEN DO;
      WRITE FILE(R) FROM(SCREEN2);
      READ FILE(R) INTO(STRUCT2);
      IF CONTROL = ' ' THEN DO;
         OPEN FILE(U) OUTPUT;
         RECORD0=STRUCT0, BY NAME;
         RECORD0=STRUCT2, BY NAME;
         WRITE FILE(U) FROM(RECORD0);
         TOTALAREA=SUM(STRUCT2.#OFAREAS);
         MAX#OFAREAS=MAX(
            STRUCT2.#OFAREAS(1),STRUCT2.#OFAREAS(2),
            STRUCT2.#OFAREAS(3),STRUCT2.#OFAREAS(4));
         MAX#OFSUBAREAS=MAX(
            STRUCT2.#OFSUBAREAS(1),STRUCT2.#OFSUBAREAS(2),
            STRUCT2.#OFSUBAREAS(3),STRUCT2.#OFSUBAREAS(4));
         SIZEOFOPTIONAL=0;
         RECORD1=STRUCT2, BY NAME;
         WRITE FILE(U) FROM(RECORD1);
         RECORD2=STRUCT2, BY NAME;
         WRITE FILE(U) FROM(RECORD2);
         ENTERED=0;
         RECORD3=STRUCT2, BY NAME;
         WRITE FILE(U) FROM(RECORD3);
         #OFSTATIONS=0;
         STATIONS(*)=0;
         WRITE FILE(U) FROM(RECORD4);
         CLOSE FILE(U) ENV(LOCK);
```

```
      END; ELSE
         RETURNFLAG='1'B;
   END; ELSE DO;
      OPEN FILE(U) INPUT;
      READ FILE(U) INTO(RECORD0);
      IF STRUCT0.SYSPASS ¬= RECORD0.SYSPASS THEN
         RETURNFLAG='1'B;
      CLOSE FILE(U);
   END;
   IF ¬ RETURNFLAG THEN DO;
      CALL COPY(EXCEPT(' ',STRUCT0.SYSID) !! '/PARAMETERS',
                MYSTA !! '/PARAMETERS');
      CALL ADDSTATION(MYSTA);
   END;
   DO WHILE( ¬ RETURNFLAG);
      NOTSPCFYFLAG='1'B;
      DO WHILE(NOTSPCFYFLAG);
         NOTSPCFYFLAG='0'B;
         WRITE FILE(R) FROM(SCREEN4);
         READ FILE(R) INTO(STRUCT4);
         IF SPCFY(1) = ESC THEN
            IF SPCFY(3) = POS(31) &
               SPCFY(4) = POS(7)  THEN TYPE=1; ELSE
            IF SPCFY(3) = POS(31) &
               SPCFY(4) = POS(9)  THEN TYPE=2; ELSE
            IF SPCFY(3) = POS(31) &
               SPCFY(4) = POS(11) THEN TYPE=3; ELSE
            IF SPCFY(3) = POS(79) &
               SPCFY(4) = POS(22) THEN RETURNFLAG='1'B;
            ELSE DO;
               SUBSTR(SCREEN4,1842,80)=BLINK!!
               'Gecersiz konumda SPCFY tusuna bastiniz.';
               NOTSPCFYFLAG='1'B;
            END;
         ELSE DO;
            SUBSTR(SCREEN4,1842,80)=BLINK!!
                  'XMIT yerine SPCFY tusunu kullaniniz.';
            NOTSPCFYFLAG='1'B;
         END;
      END;
      SUBSTR(SCREEN4,1842,80)=' ';
      IF ¬ RETURNFLAG THEN DO;
         MESSAGE=CLS!!ESC!!'"2-'!!BLINK!!
                 'Altsisteme gecis icin lutfen bekleyiniz'!!
                 ESC!!'"o7'!!DC2ETX;
         WRITE FILE(R) FROM(MESSAGE);
         CLOSE FILE(R);
         CALL SUBSYSTEM(TYPE);
      END;
   END;
END;

ADDSTATION:PROC(STA#);
   DCL STA# PIC'9999',
       FLAG BIT(1);
   OPEN FILE(U) UPDATE;
   READ FILE(U) INTO(RECORD4) INDEX(4);
```

```
   FLAG='1'B;
   DO I=1 TO 10 WHILE(FLAG);
      IF STATIONS(I)=0 THEN DO;
         #OFSTATIONS=#OFSTATIONS+1;
         STATIONS(I)=STA#;
         FLAG='0'B;
      END;
   END;
   REWRITE FILE(U) FROM(RECORD4);
   CLOSE FILE(U) ENV(LOCK);
END ADDSTATION;

DELSTATION:PROC(STA#);
   DCL STA# PIC'9999',
       FLAG BIT(1);
   OPEN FILE(U) UPDATE;
   READ FILE(U) INTO(RECORD4) INDEX(4);
   FLAG='1'B;
   DO I=1 TO 10 WHILE(FLAG);
      IF STATIONS(I)=STA# THEN DO;
         #OFSTATIONS=#OFSTATIONS-1;
         STATIONS(I)=0;
         FLAG='0'B;
      END;
   END;
   REWRITE FILE(U) FROM(RECORD4);
   CLOSE FILE(U) ENV(LOCK);
END DELSTATION;

END SYSTEM;
```

```
/*%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/
/*                                                                      */
/*      R E X   E X A M I N A T I O N   S Y S T E M                     */
/*      ===========================================                     */
/*                                                                      */
/*         WRITTEN BY MUSTAFA TUTUNCU  (SEPT 1987)                      *
/*                                                                      */
/*%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/

EXAMINATION:PROC OPTIONS(MAIN);
DCL SYSPRINT FILE STREAM OUTPUT ENV(KIND='PRINTER',
                                    MAXRECSIZE=132);
DCL STARTSYS(03) ENTRY INIT(
        ROOM_ASSIGNMENT,           /*  1 ROOM ASSIGNMENT  */
        EVAL_OF_EXAM,              /*  2 EVALUATION       */
        PRINT_OUT);                /*  3 PRINT-OUTS       */

% INCLUDE 'REX/SCREENS/DECLARE.';
% INCLUDE 'LAYOUT/PROVINCES.';
DCL U FILE RECORD ENV(KIND='DISK',MAXRECSIZE=180,AREAS=5,
                      AREASIZE=1);

DCL 1 RECORD0,
      2 SYSID             CHAR(5),
      2 SYSPASS           CHAR(5),
      2 SYSNAME           CHAR(54);
DCL 1 RECORD1,
      2 #OFCENTERS        PIC'99',
      2 #OFSESSIONS       PIC'9',
      2 SESSIONS(4),
        3 #OFAREAS        PIC'99',
        3 #OFSUBAREAS     PIC'9',
        3 EXAMDATE        CHAR(8),
        3 EXAMHOUR        CHAR(5),
      2 TOTALAREA         PIC'99',
      2 MAX#OFAREAS       PIC'99',
      2 MAX#OFSUBAREAS    PIC'9',
      2 SIZEOFOPTIONAL    PIC'999';
DCL 1 RECORD2,
      2 CENTERS(90)       PIC'99';
DCL 1 RECORD3,
      2 EXPECTED          PIC'99999',
      2 ENTERED           PIC'99999';
DCL 1 RECORD4,
      2 #OFSTATIONS       PIC'99',
      2 STATIONS(10)      PIC'9999';

ON RECORD(U);

GETUSERPARAMS:PROC(SYSID);
   DCL SYSID CHAR(*);
   TITLE(U)=SYSID !! '/PARAMETERS';
   OPEN FILE(U) INPUT;
   READ FILE(U) INTO(RECORD0);
   READ FILE(U) INTO(RECORD1);
   READ FILE(U) INTO(RECORD2);
   READ FILE(U) INTO(RECORD3);
```

```
   READ FILE(U) INTO(RECORD4);
   CLOSE FILE(U) ENV(LOCK);
END GETUSERPARAMS;

% INCLUDE 'REX/SCREENS/210.';
% INCLUDE 'REX/SCREENS/211.';
% INCLUDE 'REX/SCREENS/212.';
% INCLUDE 'REX/SCREENS/213.';
% INCLUDE 'REX/SCREENS/214.';
% INCLUDE 'REX/SCREENS/215.';
% INCLUDE 'REX/SCREENS/216.';
% INCLUDE 'REX/SCREENS/217.';
% INCLUDE 'REX/SCREENS/218.';
% INCLUDE 'REX/SCREENS/219.';
% INCLUDE 'REX/SCREENS/220.';
% INCLUDE 'REX/SCREENS/221.';
% INCLUDE 'REX/SCREENS/222.';
/*%*%*%**%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*                   FILE DECLARATIONS                    */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%**/
DCL CANDIDATE FILE RECORD ENVIRONMENT(FILETYPE=7,
                                     KIND='DISK',
                                     FRAMESIZE=8);
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*                LAYOUT OF CANDIDATE FILE              */
/*                                                      */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
DCL 1 CANDIDATE_REC              CONTROLLED,
      2 APPLICATION#             PIC'(6)9',
      2 NAME                     CHAR(12),
      2 SURNAME                  CHAR(14),
      2 FATHER                   CHAR(12),
      2 SEX                      PIC'9',
      2 BIRTH                    PIC'(6)9',
      2 CENTER                   PIC'99',
      2 AREA(#OFSESSIONS)        PIC'99',
      2 ADDRESS,
        3 LINE1                  CHAR(23),
        3 LINE2                  CHAR(23),
        3 DISTRICT               CHAR(12),
        3 PROVINCE               PIC'99',
      2 REG_VALIDITY             PIC'9',
      2 EXAM_VALIDITY            PIC'9',
      2 SESSION(#OFSESSIONS),
        3 ROOM#                  PIC'(5)9',
        3 ROOM@                  PIC'(4)9',
        3 SEAT#                  PIC'999',
        3 ARCHIVE#,
          4 BOX#                 PIC'99',
          4 BOXSEQ#              PIC'999',
        3 ATTENDANCE             PIC'9',
        3 SUBAREA(MAX#OFSUBAREAS),
          4 CORRECT              PIC'999',
          4 INCORRECT            PIC'999',
          4 RAW                  PIC'-999.V999',
          4 STANDARD             PIC'-999.V999',
        3 WEIGHTED               PIC'-999.V999',
```

```
      2 OPTIONAL                        CHAR(SIZEOFOPTIONAL);

DCL EXAM      FILE RECORD ENVIRONMENT(
                                            KIND='DISK',
                                            MAXRECSIZE=4568,
                                            BLOCKSIZE=4568,
                                            FRAMESIZE=8);
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*               LAYOUT OF EXAM FILE                        */
/*          MAXRECSIZE=4568 BLOCKSIZE=4568                  */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
DCL 1 EXAM_REC,
      2 APPLICATION#     PIC'999999',
      2 CENTER           PIC'99',
      2 SESSION(4),
        3 ROOM#          PIC'99999',
        3 ROOM@          PIC'9999',
        3 SEAT#          PIC'999',
        3 ARCHIVE#,
          4 BOX#         PIC'99',
          4 BOXSEQ#      PIC'999',
        3 READ_FLAG      PIC'9',
        3 ATTENDANCE     PIC'9',
        3 BOOKLET_TYPE   CHAR( 1),
        3 WEIGHTED       PIC'-999.V999',
        3 AREA           PIC'99',
        3 SUBAREA(5),
          4 CORRECT      PIC'999',
          4 INCORRECT    PIC'999',
          4 RAW          PIC'-999.V999',
          4 STANDARD     PIC'-999.V999',
          4 ANSWERS(200) CHAR(1);

DCL (TEMP,TEMP1,TEMP2,TEMP3,TEMP4)
              FILE RECORD ENVIRONMENT(
                                            KIND='DISK',
                                            MAXRECSIZE=45,
                                            BLOCKSIZE=180);
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*
/*                   LAYOUT OF TEMPORARY FILE
/*              MAXRECSIZE=45  BLOCKSIZE=180 FRAMESIZE=8
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*
DCL 1 TEMP_REC,
      2 APPLICATION#             PIC'999999',
      2 CENTER                   PIC'99',
      2 ROOM#                    PIC'99999',
      2 ROOM@                    PIC'9999',
      2 AREA                     PIC'99',
      2 BUILDING#                PIC'99999',
      2 BUILDING@                PIC'9999',
      2 RND#                     PIC'999999',
      2 SESSION#                 PIC'9',
      2 ARCHIVE#,
        3 BOX#                   PIC'99',
        3 BOXSEQ#                PIC'999',
      2 OUTGOING#                PIC'99999';
```

```
DCL TEMP_FILES(4) FILE INIT(TEMP1,TEMP2,TEMP3,TEMP4);

DCL GEN_ROOM FILE RECORD ENVIRONMENT(
                                        KIND='DISK',
                                        MAXRECSIZE=119,
                                        BLOCKSIZE=119,
                                        FRAMESIZE=8,
                                        TITLE='REX/FILES/ROOM.'
                                        MYUSE='IN');

DCL ROOM     FILE RECORD ENVIRONMENT(
                                        KIND='DISK',
                                        MAXRECSIZE=120,
                                        BLOCKSIZE=,
                                        FRAMESIZE=8);
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*                    LAYOUT OF ROOM FILE                              */
/*         MAXRECSIZE=120 BLOCKSIZE=120 FRAMESIZE=8                    */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
DCL 1 ROOM_REC,
      2 CENTER                    PIC'99',
      2 ROOM#                     PIC'99999',
      2 ROOM@                     PIC'9999',
      2 ROOM_NAME,
        3 LINE1                   CHAR(30),
        3 LINE2                   CHAR(30),
      2 AREA                      PIC'99',
      2 BUILDING#                 PIC'99999',
      2 BUILDING@                 PIC'9999',
      2 ROOM_CAPACITY             PIC'999',
      2 #OFASSIGNED               PIC'999',
      2 ROOM_PRIORITY             PIC'9',
      2 BUILD_PRIORITY            PIC'999',
      2 @OFROOM_DIRECTOR          PIC'9999',
      2 @OFSUPERVISOR(6)          PIC'9999';

DCL BUILDING FILE RECORD ENVIRONMENT(
                                        KIND='DISK',
                                        MAXRECSIZE=261,
                                        BLOCKSIZE=261,
                                        FRAMESIZE=8);

/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*                  LAYOUT OF BUILDING FILE                            */
/*         MAXRECSIZE=261 BLOCKSIZE=261 FRAMESIZE=8                    */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
DCL 1 BUILDING_REC,
      3 BUILDING#                 PIC'99999',
      3 BUILDING_NAME,
        4 LINE1                   CHAR(30),
        4 LINE2                   CHAR(30),
      3 SESSION(4),
        4 @OFBUILDING_DIRECTOR    PIC'9999',
        4 @OFASST_DIRECTOR        PIC'9999',
        4 @OFBUILDING_MANAGER     PIC'9999',
        4 @OFSTANDBY(6)           PIC'9999',
        4 #OFCANDIDATES           PIC'9999',
```

```
        4 #OFROOM_DIRECTOR      PIC'9999',
        4 #OFSUPERVISOR         PIC'9999',
        4 USAGE_FLAG            PIC'9'   ;

DCL ANSWER_FILE FILE RECORD ENVIRONMENT(
                                        KIND='DISK',
                                        MAXRECSIZE=40000,
                                        BLOCKSIZE=40000,
                                        AREAS=100,
                                        AREASIZE=3,
                                        FRAMESIZE=8);

DCL 1 ANSWERKEYS_REC,
      2 A_KEYS_2(2,4,10,500) CHAR(1);

DCL DISTRB   FILE RECORD ENVIRONMENT(
                                        KIND='DISK',
                                        MAXRECSIZE=200,
                                        BLOCKSIZE=200,
                                        FRAMESIZE=8);

/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*                  LAYOUT OF DISTRB FILE                              */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%**/
DCL 1 DISTRB_REC,
      2 DISTRB_DATA(4,10)       PIC'99999';

DCL EXAMOPTIC FILE RECORD ENVIRONMENT(
                                        KIND='TAPE',
                                        MAXRECSIZE=512,
                                        BLOKCKSIZE=512,
                                        MYUSE='IN',
                                        FRAMESIZE=8);
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%**/
/*         LAYOUT OF OPTICAL FILE FOR EXAM                             */
/*      MAXRECSIZE=512 BLOCKSIZE=512 FRAMESIZE=8                       */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%**/
DCL 1 EXAMOPTIC_REC,
      2 APPLICATION#            PIC'999999',
      2 SESSION                 PIC'9',
      2 AREA                    PIC'99',
      2 BOOKLET_TYPE            CHAR(1),
      2 ATTENDANCE              PIC'9',
      2 ANSWER(500)             CHAR(1),
      2 FILLER                  CHAR(1);

DCL ZIPFILE FILE RECORD ENV(
                                        KIND='DISK',
                                        MYUSE='OUT',
                                        MAXRECSIZE=15,
                                        BLOCKSIZE=420,
                                        FRAMESIZE=48,
                                        FILEKIND='JOBSYMBOL');

DCL LIB1 LIBRARY(TITLE='OBJECT/REX/LIBRARY/STARTJOB.'),
    LIB2 LIBRARY(TITLE='OBJECT/REX/LIBRARY/FILEMANAGER.');
```

```
DCL
    STARTJOB ENTRY(CHAR(*))
             RETURNS(BIT(1)) OPTIONS(LIBRARY=LIB1),
    REMOVE ENTRY(CHAR(*))
             RETURNS(BIT(1)) OPTIONS(LIBRARY=LIB2),
    COPY   ENTRY(CHAR(*),CHAR(*))
             RETURNS(BIT(1)) OPTIONS(LIBRARY=LIB2),
    CHANGE ENTRY(CHAR(*),CHAR(*))
             RETURNS(BIT(1)) OPTIONS(LIBRARY=LIB2),
    SEARCH ENTRY(CHAR(*),CHAR(*))
             RETURNS(BIT(1)) OPTIONS(LIBRARY=LIB2);

DCL MYSELF BUILTIN;
DCL MYSTA  PIC '9999';

/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*                    SCREEN LAYOUTS                                  */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
DCL 1 STRUCT999,
      2 SPCFY(4)         CHAR(1);

DCL 1 STRUCT221,
      2 WS(5)            CHAR(6);

DCL 1 STRUCT213 CONTROLLED,
      2 AREADATA(TOTALAREA) PIC '99999' ;

/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*                  DATA STRUCTURES                                   */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
DCL BT# FIXED(2);
DCL KEY_PTRS(#OFSESSIONS,MAX#OFAREAS,MAX#OFSUBAREAS)
             FIXED(3) CONTROLLED;
DCL A_KEYS  (BT#,#OFSESSIONS,MAX#OFAREAS,500)
                                          CHAR(1) CONTROLLED;
DCL ATABLE (#OFSESSIONS,#OFCENTERS,10)  FIXED(6) CONTROLLED;
DCL SUMTABLE(#OFSESSIONS,#OFCENTERS,10) FIXED(6) CONTROLLED;
DCL SPCFYOKFLAG          BIT(1) INIT('1'B);
DCL TYPE                 FIXED;
DCL MSG                  CHAR(500) VAR;

/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*                         PROCEDURES                                 */
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
WRTOZIP:PROC(JOBRECORD,JOBSEQUENCE);
   DCL JOBRECORD CHAR(*),
       JOBSEQUENCE FIXED;
   DCL 1 JOBREC,
         2 JOBDATA CHAR(82),
         2 JOBSEQ  PIC '99999999';
   JOBDATA=JOBRECORD;
   JOBSEQ =JOBSEQUENCE;
   WRITE FILE(ZIPFILE) FROM(JOBREC);
END WRTOZIP;

/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*/
/*              REX EXAMINATION SUBSYSTEMS                            */
```

```
/*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%**/
ROOM_ASSIGNMENT:PROCEDURE;

   GET_DISTRB:PROCEDURE;
/*%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/
/* GET DISTRIBUTION PROCESS OF ROOM ASSIGNMENT SUBSYSTEM  */
/*%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/
   DCL SPCFYOKFLAG                BIT(1) INIT('1'B);
   DCL TYPE                       FIXED;
   DCL OFFSET                     FIXED;
DO WHILE(TYPE ¬= 99);
DO WHILE(SPCFYOKFLAG);
   SPCFYOKFLAG='0'B;
               WRITE FILE(R) FROM(SCREEN212);
               READ  FILE(R) INTO(STRUCT999);
   IF SPCFY(1) = ESC THEN DO;
      IF ( SPCFY(3) = POS(15) ! SPCFY(3) = POS(79))&
         ( SPCFY(4) = POS(6)  ! SPCFY(4) = POS(8)   !
           SPCFY(4) = POS(22) )
      THEN DO;
           TYPE=INDEX(ASCII,SPCFY(4));
           TYPE=TYPE/2-2;
           IF SPCFY(3) = POS(79) THEN TYPE=99;
      END;
      ELSE TYPE=0;
      IF TYPE = 0 THEN DO;
         SUBSTR(SCREEN212,1842,80)=BLINK!!
              'Gecersiz konumda SPCFY tusuna bastiniz.';
         SPCFYOKFLAG='1'B;
      END;
   END;
   ELSE DO;
      SUBSTR(SCREEN212,1842,80)=BLINK!!
            'XMIT yerine SPCFY tusunu kullaniniz.';
      SPCFYOKFLAG='1'B;
   END;
END;
SUBSTR(SCREEN212,1842,80)=' ';
IF TYPE ¬= 99 THEN
   IF TYPE = 1 THEN DO;/* DISTRIBUTION FROM SCREEN*/
      IF #OFSESSIONS < 4 THEN  DO;
         X=(4*#OFSESSIONS+5)*80+2;
         Y=1280-320*#OFSESSIONS;
         SUBSTR(SCREEN213,X,Y)=' ';
      END;
      DO I=1 TO #OFSESSIONS;
         IF #OFAREAS(I) < 10 THEN DO;
           X=320*I+240+8*#OFAREAS(I)+3;
           Y=79-8*#OFAREAS(I);
           SUBSTR(SCREEN213,X,Y)=' ';
           X=X-80;
           SUBSTR(SCREEN213,X,Y)=' ';
         END;
       END;
       SUBSTR(SCREEN213,1842,80)=' ';
       SUBSTR(SCREEN213,1762,80)=' ';
       DO I=1 TO #OFCENTERS;
```

```
          SUBSTR(SCREEN213,259,2)=CENTERS(I);
          ON CONVERSION BEGIN;
        SUBSTR(SCREEN213,1842,80)=BRIGHT !! BLINK !!
        'Dagilimi girip SPCFY Yerine XMIT tusuna basiniz';
        GO LW;
       END;
       LW:WRITE FILE(R) FROM(SCREEN213);
          READ  FILE(R) INTO(STRUCT213);
          OFFSET=0;
          DO J=1 TO #OFSESSIONS;
               DO K=1 TO #OFAREAS(J);
               ATABLE(J,I,K)=AREADATA(K+OFFSET);
               END;
               OFFSET=OFFSET+#OFAREAS(J);
          END;
      END;
         TITLE(DISTRB)=SYSID !! '/DISTRB.';
         OPEN FILE(DISTRB) OUTPUT;
         DO II=1 TO #OFCENTERS;
             DO JJ=1 TO #OFSESSIONS;
                 DO KK=1 TO 10;
                    DISTRB_DATA(JJ,KK)=ATABLE(JJ,II,KK);
                 END;
             END;
             WRITE FILE(DISTRB) FROM(DISTRB_REC);
         END;
         CLOSE FILE(DISTRB) ENV(LOCK);
         SUBSTR(SCREEN212,1762,80)=
               BRIGHT !! BLINK !! REVERSE !!
         SYSID!!'/DISTRB '!!NORMAL!!BRIGHT!!BLINK!!
         'adli dagilim k~t~g~ yaratildi.';
         SUBSTR(SCREEN212,1842,80)=BRIGHT !!
        'Geri donus icin GERI DONUS secenegini kullaniniz';
   END;ELSE IF TYPE = 2 THEN DO;/* DISTRIBUTION FROM FILE*/
       TITLE(DISTRB)=SYSID !! '/DISTRB.';
       OPEN FILE(DISTRB) INPUT;
          DO II=1 TO #OFCENTERS;
              READ FILE(DISTRB) INTO(DISTRB_REC);
              DO JJ=1 TO #OFSESSIONS;
                  DO KK=1 TO 10;
                     ATABLE(JJ,II,KK)=DISTRB_DATA(JJ,KK);
                  END;
              END;
          END;
          CLOSE FILE(DISTRB) ENV(LOCK);
          SUBSTR(SCREEN212,1762,80)=BRIGHT !! BLINK !!
          'Aday dagilimi ' !! REVERSE !!
          SYSID !! '/DISTRB ' !! NORMAL !! BRIGHT!!BLINK!!
          'adli dagilim k~t~g~nden okundu.';
          SUBSTR(SCREEN212,1842,80)=BRIGHT !!
        'Geri donus icin GERI DONUS secenegini kullaniniz';
   END ;
SPCFYOKFLAG='1'B;
END;
   SUBSTR(SCREEN212,1762,160)=' ';
/* DO I=1 TO #OFCENTERS;
      PUT SKIP EDIT('MERKEZ:',CENTERS(I))(A,F(2),SKIP);
```

```
      DO J=1 TO #OFSESSIONS;
        L=#OFAREAS(J);
        PUT SKIP EDIT(J,(ATABLE(J,I,K) DO K=1 TO L))
                     (F(2),X(2),(L)(F(5),X(1)));
      END;
   END; DISPLAY('BOS ACCEPT GIRINIZ') REPLY(I);  */
END GET_DISTRB;
/*************************************************************/
CREATE_FILES:PROCEDURE;

CR_ROOM_BUILD_FILES:PROCEDURE;
/*%#%#%##%#%#%##%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/
/*                  CREATION OF FILES                          */
/*%#%#%##%#%#%##%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/
   DCL W@ PIC'9999';
   DCL OLDBUILDING@ PIC'9999';
   DCL (CEN,ERA) FIXED;
   ALLOCATE SUMTABLE;
   MSG='Salon ve Bina k~t~kleri yaratiliyor..'!!
        REPEAT(' ',81)!!CR!!ETX;
   WRITE FILE(R) FROM(MSG);
   CALL COPY('REX/FILES/BUILDING',SYSID !! '/BUILDING');
   TITLE(BUILDING)= SYSID !! '/BUILDING';
   OPEN FILE(BUILDING) UPDATE;
   OPEN FILE(GEN_ROOM);
   ON ERROR BEGIN;
      PUT DATA(#OFSESSIONS,#OFAREAS,CENTERS(1),CENTERS(2),
               ICHAR,CEN,ERA,SUMTABLE);
      STOP;
   END;
   DO ICHAR=1 TO #OFSESSIONS;
      MSG=SYSID!!'/ROOM/'!!ICHAR!!
                ' Adli salon kutugu yaratildi.'
      !!REPEAT(' ',81)!!CR!!LF!!ETX;
      TITLE(ROOM) = SYSID !! '/ROOM/' !! ICHAR;
      OPEN FILE(ROOM) OUTPUT;
      OLDBUILDING@=9999;
      W@=0;
      CEN=1;
      ERA=1;
      GET_REC:READ FILE(GEN_ROOM) INTO(ROOM_REC);
      IF CEN <= #OFCENTERS THEN DO;
         IF CENTERS(CEN) = ROOM_REC.CENTER THEN DO;
            IF SUMTABLE(ICHAR,CEN,ERA) >
               ATABLE(ICHAR,CEN,ERA)
            THEN DO;
               ERA=ERA+1;
               IF ERA > #OFAREAS(ICHAR) THEN DO;
                  CEN=CEN+1;
                  ERA=1;
                  GO TO GET_REC;
               END;ELSE;
            END;ELSE;
         END;ELSE GO TO GET_REC;
      END;ELSE GOTO ALL_OK;
      ROOM_REC.AREA=ERA;
      W@=W@+1;
```

```
      ROOM_REC.ROOM@=W@;
      IF CEN <= #OFCENTERS THEN DO;
      WRITE FILE(ROOM) FROM(ROOM_REC);
      SUMTABLE(ICHAR,CEN,ERA)=SUMTABLE(ICHAR,CEN,ERA)+
                              ROOM_REC.ROOM_CAPACITY;
      END;ELSE GO TO GET_REC;
      IF OLDBUILDING@ ¬= ROOM_REC.BUILDING@ THEN DO;
         OLDBUILDING@=ROOM_REC.BUILDING@;
         READ FILE(BUILDING) INTO(BUILDING_REC)
                             INDEX(ROOM_REC.BUILDING@);
         BUILDING_REC.SESSION(ICHAR).USAGE_FLAG = 1;
         WRITE FILE(BUILDING) FROM(BUILDING_REC)
                              INDEX(ROOM_REC.BUILDING@);
      END;
      GO TO GET_REC;
      ALL_OK: CLOSE FILE(ROOM)     ENVIRONMENT(CRUNCH);
              CLOSE FILE(GEN_ROOM) ENVIRONMENT(LOCK);
              CLOSE FILE(BUILDING) ENVIRONMENT(LOCK);
              OPEN  FILE(BUILDING) UPDATE;
              OPEN  FILE(GEN_ROOM);
      WRITE FILE(R) FROM(MSG);
    END;
    CLOSE FILE(BUILDING);
    MSG=SYSID!!'/BUILDING'!!' Adli bina k~t~g~ yaratildi.'!!
        REPEAT(' ',81)!!CR!!ETX;
    WRITE FILE(R) FROM(MSG);
    MSG='?DEVAM ETMEK I@IN SPCFY TUSUNA BASINIZ'!!
        REPEAT(' ',81)!!CR!!ETX;
    WRITE FILE(R) FROM(MSG);
    READ FILE(R) INTO(STRUCT999);
/* DO I=1 TO #OFCENTERS;
      PUT SKIP EDIT('MERKEZ:',CENTERS(I))(A,F(2),SKIP);
      DO J=1 TO #OFSESSIONS;
        L=#OFAREAS(J);
        PUT SKIP EDIT(J,(SUMTABLE(J,I,K) DO K=1 TO L))
                      (F(2),X(2),(L)(F(5),X(1)));
      END;
   END;DISPLAY('BOS ACCEPT GIRINIZ') REPLY(I);  */
END CR_ROOM_BUILD_FILES;

CR_EXAM_FILE:PROCEDURE;
   DCL TRUE     BIT(1) INIT('1'B);
   TITLE(CANDIDATE) = SYSID !! '/CANDIDATE';
   TITLE(EXAM)      = SYSID !! '/EXAMINATION';
   OPEN FILE(CANDIDATE) INPUT;
   OPEN FILE(EXAM)      OUTPUT;
   ON ENDFILE(CANDIDATE) BEGIN;
      CLOSE FILE(CANDIDATE) ENV(LOCK);
      CLOSE FILE(EXAM)      ENV(LOCK);
      GO XIT;
   END;
   MSG=SYSID !!
       '/EXAMINATION adli sinav k~t~g~ yaratiliyor.'!!
            REPEAT(' ',81)!!CR!!ETX;
   WRITE FILE(R) FROM(MSG);
         EXAM_REC.ROOM#=0;
         EXAM_REC.ROOM@=0;
```

```
         EXAM_REC.SEAT#=0;
         EXAM_REC.BOX#=0;
         EXAM_REC.BOXSEQ#=0;
         EXAM_REC.READ_FLAG=0;
         EXAM_REC.ATTENDANCE=0;
         EXAM_REC.BOOKLET_TYPE=' ';
         EXAM_REC.WEIGHTED=0;
         EXAM_REC.AREA=0;
         EXAM_REC.CORRECT=0;EXAM_REC.INCORRECT=0;
         EXAM_REC.RAW=0;
         EXAM_REC.STANDARD=0;EXAM_REC.ANSWERS=' ';
   DO WHILE(TRUE);
         READ FILE(CANDIDATE) INTO(CANDIDATE_REC);
         EXAM_REC.APPLICATION# = CANDIDATE_REC.APPLICATION#;
         EXAM_REC.CENTER       = CANDIDATE_REC.CENTER;
         WRITE FILE(EXAM) FROM(EXAM_REC);

   END;
   XIT:
   MSG='Sinav k~t~g~ yaratildi...'!!
       REPEAT(' ',81)!!CR!!ETX;WRITE FILE(R) FROM(MSG);
   MSG='?DEVAM ETMEK I@IN SPCFY TUSUNA BASINIZ'!!
       REPEAT(' ',81)!!CR!!ETX;
   WRITE FILE(R) FROM(MSG);READ FILE(R) INTO(STRUCT999);
END CR_EXAM_FILE;

CR_TEMP_FILES:PROCEDURE;
   DCL TRUE     BIT(1) INIT('1'B);
   DCL RNDNUM   FLOAT;
   DCL ICHAR    PIC'9';
   DCL TTTT     CHAR(20);
   CHANGE2:PROCEDURE(FNAME,FTITLE);
      DCL FNAME FILE;
      DCL FTITLE CHAR(*) VAR;
          TITLE(FNAME) = FTITLE;
   END CHANGE2;
   TITLE(CANDIDATE) = SYSID !! '/CANDIDATE';
   OPEN FILE(CANDIDATE) INPUT;
   DO ICHAR=1 TO #OFSESSIONS;
      TTTT = SYSID !! '/TEMP/' !! ICHAR;
      CALL CHANGE2(TEMP_FILES(ICHAR),TTTT);
      OPEN FILE(TEMP_FILES(ICHAR)) OUTPUT;
   END;
   ON ENDFILE(CANDIDATE) BEGIN;
      DO ICHAR=1 TO #OFSESSIONS;
         CLOSE FILE(TEMP_FILES(ICHAR)) ENV(LOCK);
      END;
      CLOSE FILE(CANDIDATE) ENV(LOCK);
      GO TO XIT;
   END;
   MSG='Ara k~t~kler yaratiliyor'!!REPEAT(' ',81)!!CR!!ETX;
   WRITE FILE(R) FROM(MSG);
   DO WHILE(TRUE);
      READ FILE(CANDIDATE) INTO(CANDIDATE_REC);
      TEMP_REC.APPLICATION# = CANDIDATE_REC.APPLICATION#;
      TEMP_REC.CENTER       = CANDIDATE_REC.CENTER;
      TEMP_REC.RND# = RANDOM(RNDSYS)*1000000;
```

```
      DO ICHAR=1 TO #OFSESSIONS;
         IF CANDIDATE_REC.AREA(ICHAR) ¬= 0 THEN DO;
            TEMP_REC.SESSION# = ICHAR;
            TEMP_REC.AREA     = CANDIDATE_REC.AREA(ICHAR);
            TEMP_REC.ROOM#=0;
            TEMP_REC.ROOM@=0;
            TEMP_REC.BUILDING#=0;
            TEMP_REC.BUILDING@=0;
            TEMP_REC.BOX#=0;
            TEMP_REC.BOXSEQ#=0;
            TEMP_REC.OUTGOING#=0;
            WRITE FILE(TEMP_FILES(ICHAR)) FROM(TEMP_REC);
         END;
      END;
   END;
   XIT:
   MSG='Ara k~t~kler yaratildi...'!!REPEAT(' ',81)!!CR!!ETX;
   WRITE FILE(R) FROM(MSG);
  MSG='?DEVAM ETMEK I@IN SPCFY TUSUNA BASINIZ'!!
      REPEAT(' ',81)!!CR!!ETX;
   WRITE FILE(R) FROM(MSG);READ FILE(R) INTO(STRUCT999);
END CR_TEMP_FILES;
DCL SPCFYOKFLAG          BIT(1) INIT('1'B);
DCL TYPE                 FIXED;
DO WHILE(TYPE ¬= 99);
DO WHILE(SPCFYOKFLAG);
   SPCFYOKFLAG='0'B;
   WRITE FILE(R) FROM(SCREEN214);
   READ  FILE(R) INTO(STRUCT999);
   IF SPCFY(1) = ESC THEN DO;
      IF ( SPCFY(3) = POS(15) ! SPCFY(3) = POS(79))&
         ( SPCFY(4) = POS(6)  ! SPCFY(4) = POS(8)   !
           SPCFY(4) = POS(10) !
           SPCFY(4) = POS(22) )
      THEN DO;
            TYPE=INDEX(ASCII,SPCFY(4));
           TYPE=TYPE/2-2;
           IF SPCFY(3) = POS(79) THEN TYPE=99;
      END;
      ELSE TYPE=0;
      IF TYPE = 0 THEN DO;
         SUBSTR(SCREEN214,1842,80)=BLINK!!
               'Gecersiz konumda SPCFY tusuna bastiniz.';
         SPCFYOKFLAG='1'B;
      END;
   END;
   ELSE DO;
      SUBSTR(SCREEN214,1842,80)=BLINK!!
            'XMIT yerine SPCFY tusunu kullaniniz.';
      SPCFYOKFLAG='1'B;
   END;
END;
SUBSTR(SCREEN214,1842,80)=' ';
IF TYPE ¬= 99 THEN
    /* DISPLAY(TYPE);     */
   IF TYPE = 1 THEN CALL CR_ROOM_BUILD_FILES;ELSE;
   IF TYPE = 2 THEN CALL CR_EXAM_FILE;ELSE;
```

```
   IF TYPE = 3 THEN CALL CR_TEMP_FILES;
SPCFYOKFLAG='1'B;
END;
END CREATE_FILES;

ASSIGN_CANDIDATES:PROCEDURE;
DCL #OFMAN PIC'99999';
SORT_FILES:PROC;
/********************/
DCL ZIPTITLE CHAR(64) VAR;
DCL ICHAR    PIC '9';
DCL JSEQ   FIXED;
   ZIPTITLE=MYSTA !! '/ZIPFILE';
   TITLE(ZIPFILE)=ZIPTITLE !! '.';
   OPEN FILE(ZIPFILE);
   CALL WRTOZIP('?BEGIN JOB;USER=REX/REX',1);
   CALL WRTOZIP('?STATION=' !! MYSTA,2);
   CALL WRTOZIP(
   '?FILE D(DISK,TITLE='!!MYSTA!!'/WAIT2)',3);
   DO ICHAR=1 TO #OFSESSIONS;  JSEQ=3+10*(ICHAR-1);
   CALL WRTOZIP(
   '?COMPILE SORTTEMP'!!ICHAR!!' SORT GO;',JSEQ+1);
   CALL WRTOZIP('?SORT DATA',JSEQ+2);
   CALL WRTOZIP(
   '   FILE IN (TITLE="'!!SYSID!!'/TEMP/'!!ICHAR!!'")',
       JSEQ+3);
   CALL WRTOZIP(
   '   FILE OUT(TITLE="'!!SYSID!!'/TEMP/'!!ICHAR !! '")',
       JSEQ+4);
   CALL WRTOZIP('   KEY(7 2)(18 2)(1 6)',JSEQ+5);
   CALL WRTOZIP(
   '?COMPILE SORTROOM'!!ICHAR!!' SORT GO;',JSEQ+6);
   CALL WRTOZIP('?SORT DATA',JSEQ+7);
   CALL WRTOZIP(
   '   FILE IN (TITLE="'!!SYSID!!'/ROOM/'!!ICHAR!!'")',
       JSEQ+8);
   CALL WRTOZIP(
   '   FILE OUT(TITLE="'!!SYSID!!'/ROOM/'!!ICHAR !! '")',
       JSEQ+9);
   CALL WRTOZIP('   KEY(1 2)(72 2)(83 3)(3 5)',JSEQ+10);
   END;
   CALL WRTOZIP('?D(MYUSE=OUT);OPEN(D);LOCK(D)',JSEQ+11);
   CALL WRTOZIP('?END JOB',JSEQ+12);
   CLOSE FILE(ZIPFILE) ENV(LOCK);
   CALL STARTJOB(ZIPTITLE);
   CALL REMOVE(ZIPTITLE);
END SORT_FILES;
/*-------------------------------------------------------*/
DISPLAY('K~t~kler siralaniyor...');
    CALL SORT_FILES;    DISPLAY('Salon atama basladi....');
    ON ERROR BEGIN;
       PUT DATA(ROOMKEY);
       STOP;
    END;
DO ICHAR=1 TO #OFSESSIONS;
   #OFMAN=0;
   DISPLAY('Oturum:' !! ICHAR);
```

```
TITLE(TEMP)=SYSID !! '/TEMP/' !! ICHAR;
TITLE(ROOM)=SYSID !! '/ROOM/' !! ICHAR;
 TITLE(EXAM)=SYSID !! '/EXAMINATION.' ;
OPEN FILE(TEMP) UPDATE;
OPEN FILE(ROOM) UPDATE;
OPEN FILE(EXAM) UPDATE;
BOTEMP:
  ON ENDFILE(TEMP) BEGIN;
     DISPLAY('TEMP BITTI');
     REWRITE FILE(ROOM) FROM(ROOM_REC);
     GO TO XIT;
  END;
  READ FILE(TEMP) INTO(TEMP_REC);
  #OFMAN=#OFMAN+1;
  IF TEMP_REC.SESSION# ¬= ICHAR THEN DO;
     DISPLAY(TEMP_REC.APPLICATION#);
     DISPLAY('ADAY KUTUKTE OLMALIYDI--');
     DISPLAY(ICHAR);
     GO TO BOTEMP;
  END;
BOCENTER:
  ON ENDFILE(ROOM) BEGIN;
     DISPLAY('SALON BITTI');
     GO TO XIT;
  END;
  READ FILE(ROOM) INTO(ROOM_REC);
  IF ROOM_REC.CENTER < TEMP_REC.CENTER
                      THEN GO TO BOCENTER;
  IF ROOM_REC.AREA ¬= TEMP_REC.AREA THEN
                      THEN GO TO BOCENTER;
  IF ROOM_REC.#OFASSIGNED >=
     ROOM_REC.ROOM_CAPACITY THEN GO TO BOCENTER;
  IF ROOM_REC.CENTER = TEMP_REC.CENTER
                           THEN GO TO DOASSIGN;
FINDMEN:
  READ FILE(TEMP) INTO(TEMP_REC);
  #OFMAN=#OFMAN+1;
  IF ROOM_REC.CENTER>TEMP_REC.CENTER THEN GO TO FINDMEN;
  IF ROOM_REC.CENTER<TEMP_REC.CENTER
                           THEN GO TO BOCENTER;
DOASSIGN:
  ROOMKEY=FLOOR(TEMP_REC.APPLICATION#/10) - 1;
  READ FILE(EXAM) INTO(EXAM_REC) INDEX(ROOMKEY);
  ROOM_REC.#OFASSIGNED=ROOM_REC.#OFASSIGNED+1;
  EXAM_REC.SESSION.ROOM#(ICHAR)=ROOM_REC.ROOM#;
  EXAM_REC.SESSION.ROOM@(ICHAR)=ROOM_REC.ROOM@;
  EXAM_REC.SESSION.SEAT#(ICHAR)=ROOM_REC.#OFASSIGNED;
  REWRITE FILE(EXAM) FROM(EXAM_REC);
  TEMP_REC=ROOM_REC,BY NAME;
  REWRITE FILE(TEMP) FROM(TEMP_REC);
  IF ROOM_REC.#OFASSIGNED <
     ROOM_REC.ROOM_CAPACITY THEN GO TO READCANDT;
  REWRITE FILE(ROOM) FROM(ROOM_REC);
  GO TO BOTEMP;
READROOM:
  READ FILE(ROOM) INTO(ROOM_REC);
  IF ROOM_REC.CENTER < TEMP_REC.CENTER
```

```
                              THEN GO TO READROOM;
      IF ROOM_REC.AREA ¬= TEMP_REC.AREA
                              THEN GO TO READROOM;
      IF ROOM_REC.#OFASSIGNED >=
         ROOM_REC.ROOM_CAPACITY THEN GO TO READROOM;
      IF ROOM_REC.CENTER ¬= TEMP_REC.CENTER
                                 THEN GO TO READROOM;
      GO TO DOASSIGN;
    READCANDT:
      READ FILE(TEMP) INTO(TEMP_REC);
      #OFMAN=#OFMAN+1;
      IF ROOM_REC.CENTER < TEMP_REC.CENTER THEN DO;
         REWRITE FILE(ROOM) FROM(ROOM_REC);
         GO TO READROOM;
      END;
      IF ROOM_REC.AREA > TEMP_REC.AREA THEN DO;
         DISPLAY('ADAY ATANAMADI');
         GO TO READCANDT;
      END;
      IF ROOM_REC.AREA ¬= TEMP_REC.AREA THEN DO;
         REWRITE FILE(ROOM) FROM(ROOM_REC);
         GO TO READROOM;
      END;
      IF ROOM_REC.CENTER = TEMP_REC.CENTER
                              THEN GO TO DOASSIGN;
      REWRITE FILE(ROOM) FROM(ROOM_REC);
      GO TO BOCENTER;
    XIT: DISPLAY(ICHAR!!'. Oturumda atanan aday sayisi:'
                 !!#OFMAN);
      CLOSE FILE(ROOM) ENV(LOCK);
      CLOSE FILE(TEMP) ENV(LOCK);
END;
      CLOSE FILE(EXAM) ENV(LOCK);
      DISPLAY('Salon atama bitti...');
END ASSIGN_CANDIDATES;

SET_ARCHIVE_NO:PROCEDURE;
    DCL WBOX#     PIC'99';
    DCL WBOXSEQ#  PIC'999';
    DCL OLDCENTER PIC'99';
    DCL ICHAR     PIC '9';
    DCL TRUE BIT(1) INIT('1'B);
    MSG=CLS;
    WRITE FILE(R) FROM(MSG);
    TITLE(EXAM)=SYSID!!'/EXAMINATION';
    OPEN FILE(EXAM) UPDATE;
    /*-------------------------------------------------------*
     *              GENERATING ARCHIVE NUMBERS               *
     *-------------------------------------------------------*/
     DO ICHAR=1 TO #OFSESSIONS;
        TITLE(TEMP)=SYSID !! '/TEMP/' !! ICHAR;
        MSG=ICHAR !! '. ARAKUTUK SIRALANIYOR'!!CR!!LF!!ETX;
        WRITE FILE(R) FROM(MSG);
        SORT TEMP_REC ON ASCENDING KEY
             (TEMP_REC.CENTER,
              TEMP_REC.ROOM#,
              TEMP_REC.APPLICATION#);
```

```
   MSG=ICHAR !! '. ARAKUTUK SIRALANDI' !! CR !!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   ON ENDFILE(TEMP) BEGIN;
      MSG=
      ICHAR!!'. ARAKUTUGE ARSIV KATSAYILARI ISLENDI'!!CR
          !!LF!!ETX;WRITE FILE(R) FROM(MSG) ;
      CLOSE FILE(TEMP) ENV(LOCK);
      GO TO XIT;
   END;
   OPEN FILE(TEMP) UPDATE;
   MSG=ICHAR
       !!'. ARAKUTUGE ARSIV KATSAYILARI ISLENIYOR.'
       !!CR!!LF;
       !!ETX; WRITE FILE(R) FROM(MSG);
   OLDCENTER=0;
   WBOX#=1;
   WBOXSEQ#=1;
   DO WHILE(TRUE);
      READ FILE(TEMP) INTO(TEMP_REC);
      IF TEMP_REC.CENTER ¬= OLDCENTER THEN DO;
         WBOX#=1;
         WBOXSEQ#=1;
         OLDCENTER=TEMP_REC.CENTER;
      END;
      TEMP_REC.BOX#=WBOX#;
      TEMP_REC.BOXSEQ#=WBOXSEQ#;
      REWRITE FILE(TEMP) FROM(TEMP_REC);
      READ FILE(EXAM) INTO(EXAM_REC)
           INDEX(FLOOR(TEMP_REC.APPLICATION#/10)-1);
      EXAM_REC.BOX#(ICHAR)=WBOX#;
      EXAM_REC.BOXSEQ#(ICHAR)=WBOXSEQ#;
      REWRITE FILE(EXAM) FROM(EXAM_REC);
      IF WBOXSEQ#=999 THEN DO;
         WBOX#=WBOX#+1;
         WBOXSEQ#=1;
      END;ELSE
         WBOXSEQ#=WBOXSEQ#+1;
   END;
  XIT:END;
  MSG='KATSAYILAR ISLENDI DEVAM ICIN?SPCFY?TUSUNA BASIN'
      !!'IZ,'!!CR!!LF!!ETX;WRITE FILE(R) FROM(MSG);
  READ FILE(R) INTO(STRUCT999);
/*-------------------------------------------------------
 *             GENERATING OUTGOING NUMBERS               *
 *-------------------------------------------------------*/
 DO ICHAR=1 TO #OFSESSIONS;
    TITLE(TEMP)=SYSID !! '/TEMP/' !! ICHAR;
    MSG=ICHAR !! '. ARAKUTUK SIRALANIYOR'!!CR!!LF!!ETX;
    WRITE FILE(R) FROM(MSG);
    SORT TEMP_REC ON ASCENDING KEY
         (TEMP_REC.CENTER,
          TEMP_REC.BUILDING#,
          TEMP_REC.AREA,
          TEMP_REC.ROOM#,
          TEMP_REC.APPLICATION#);
    MSG=ICHAR !! '. ARAKUTUK SIRALANDI' !!CR!!LF!!ETX;
    WRITE FILE(R) FROM(MSG);
```

```
      ON ENDFILE(TEMP) BEGIN;
         MSG=
         ICHAR!!'. ARAKUTUGE ARSIV KATSAYILARI ISLENDI'
         !!CR!!LF!!ETX;WRITE FILE(R) FROM(MSG) ;
         CLOSE FILE(TEMP) ENV(LOCK);
         GO TO XIT2;
      END;
      OPEN FILE(TEMP) UPDATE;
      MSG=ICHAR!!'. ARAKUTUGE GIDIS SIRA NOLAR ISLENIYOR.'
         !!CR!!LF!!ETX; WRITE FILE(R) FROM(MSG);
      OLDCENTER=0;
      WBOX#=1;
      WBOXSEQ#=1;
      DO WHILE(TRUE);
         READ FILE(TEMP) INTO(TEMP_REC);
         IF TEMP_REC.CENTER ¬= OLDCENTER THEN DO;
            WBOX#=1;
            WBOXSEQ#=1;
            OLDCENTER=TEMP_REC.CENTER;
         END;
         TEMP_REC.OUTGOING#=1000 * WBOX# + WBOXSEQ#;
         REWRITE FILE(TEMP) FROM(TEMP_REC);
         IF WBOXSEQ#=999 THEN DO;
            WBOX#=WBOX#+1;
            WBOXSEQ#=1;
         END;ELSE
            WBOXSEQ#=WBOXSEQ#+1;
      END;
   XIT2:END;
   MSG='GIDIS KATSAYILARI ISLENDI '!!
       ' GERI DONUS ICIN?SPCFY?TUSUNA BASINIZ.'
       !!CR!!LF!!ETX;WRITE FILE(R) FROM(MSG);
   READ FILE(R) INTO(STRUCT999);
END SET_ARCHIVE_NO;

TRANS_ASSIGNMENT_DATA:PROCEDURE;
   DCL TRUE BIT(1) INIT('1'B);
   DCL (I,J,K) FIXED;
   TITLE(CANDIDATE)=SYSID !! '/CANDIDATE';
   TITLE(EXAM)=     SYSID !! '/EXAMINATION';
   OPEN FILE(CANDIDATE) UPDATE;
   OPEN FILE(EXAM) INPUT;
   ON ENDFILE(EXAM) BEGIN;
      MSG='Atama bilgileri Aday kutugune aktarildi.'
          !!CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
      CLOSE FILE(CANDIDATE) ENV(LOCK);
      CLOSE FILE(EXAM);
      MSG='DEVAM ETMEK ICIN?SPCFY?TUSUNA BASINIZ.'
          !!CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
      READ FILE(R) INTO(STRUCT999);
      GO TO XIT;
   END;
   MSG='Atama bilgileri Aday kutugune aktariliyor'
        !!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
```

```
   DO WHILE(TRUE);
      READ FILE(EXAM) INTO(EXAM_REC);
      READ FILE(CANDIDATE) INTO(CANDIDATE_REC)
           INDEX(FLOOR(EXAM_REC.APPLICATION#/10)-1);
      DO I=1 TO #OFSESSIONS;
         CANDIDATE_REC.ROOM#(I)=EXAM_REC.ROOM#(I);
         CANDIDATE_REC.ROOM@(I)=EXAM_REC.ROOM@(I);
         CANDIDATE_REC.SEAT#(I)=EXAM_REC.SEAT#(I);
         CANDIDATE_REC.BOX#(I)=EXAM_REC.BOX#(I);
         CANDIDATE_REC.BOXSE@#(I)=EXAM_REC.BOXSE@#(I);
      END;
      REWRITE FILE(CANDIDATE) FROM(CANDIDATE_REC);
   END;
   XIT:
END TRANS_ASSIGNMENT_DATA;

DCL SPCFYOKFLAG          BIT(1) INIT('1'B);
DCL TYPE                 FIXED;
DO WHILE(TYPE ¬= 99);
DO WHILE(SPCFYOKFLAG);
   SPCFYOKFLAG='0'B;
   WRITE FILE(R) FROM(SCREEN211);
   READ  FILE(R) INTO(STRUCT999);
   IF SPCFY(1) = ESC THEN DO;
      IF ( SPCFY(3) = POS(15) ! SPCFY(3) = POS(79))&
         ( SPCFY(4) = POS(6)  ! SPCFY(4) = POS(8)   !
           SPCFY(4) = POS(10) ! SPCFY(4) = POS(12)  !
            SPCFY(4) = POS(14) ! SPCFY(4) = POS(22) )
      THEN DO;
           TYPE=INDEX(ASCII,SPCFY(4));
           TYPE=TYPE/2-2;
           IF SPCFY(3) = POS(79) THEN TYPE=99;
      END;
      ELSE TYPE=0;
      IF TYPE = 0 THEN DO;
         SUBSTR(SCREEN211,1842,80)=BLINK!!
               'Gecersiz konumda SPCFY tusuna bastiniz.';
         SPCFYOKFLAG='1'B;
      END;
   END;
   ELSE DO;
      SUBSTR(SCREEN211,1842,80)=BLINK!!
            'XMIT yerine SPCFY tusunu kullaniniz.';
      SPCFYOKFLAG='1'B;
   END;
END;
SUBSTR(SCREEN211,1842,80)=' ';
IF TYPE ¬= 99 THEN
    /* DISPLAY(TYPE);     */
   IF TYPE = 1 THEN CALL GET_DISTRB;ELSE
   IF TYPE = 2 THEN CALL CREATE_FILES;ELSE
   IF TYPE = 3 THEN CALL ASSIGN_CANDIDATES;ELSE
   IF TYPE = 4 THEN CALL SET_ARCHIVE_NO;ELSE
   IF TYPE = 5 THEN CALL TRANS_ASSIGNMENT_DATA;
SPCFYOKFLAG='1'B;
END;
END ROOM_ASSIGNMENT;
```

```
EVAL_OF_EXAM:PROCEDURE;
        GET_ANSWER_KEYS:PROCEDURE;
           DCL SAKEYS CHAR(200) VAR;
           DCL ERAKEYS CHAR(500) VAR;
           DCL 1 BTFLAG CHAR(5) VAR;
           DCL (T#,T1#) FIXED;
           DCL 1 DUMMY_PTRS(T#) PIC '999' CONTROLLED;
           DCL I PIC'9',J PIC '99',K PIC'9',L PIC '9';
           DO I=1 TO #OFSESSIONS;
              SUBSTR(SCREEN216,561,800)=' ';
              SUBSTR(SCREEN216,1840,80)=' ';
              T#=0;
              DO J=1 TO #OFAREAS(I);
               SUBSTR(SCREEN216,(J+6)*80+5,2)=J;
               DO K=1 TO #OFSUBAREAS(I);
                SUBSTR(SCREEN216,(J+6)*80+16+(K-1)*9,9)=K!!'
                      !!':?   ?   ';
                T#=T#+1;
               END;
              END;
              SUBSTR(SCREEN216,250,8)=' : ' !! I !! '    ';
           L1:WRITE FILE(R) FROM(SCREEN216);
              ALLOCATE DUMMY_PTRS;
              ON CONVERSION BEGIN;
                 SUBSTR(SCREEN216,1841,80)=
                 '???SPCFY yerine XMIT tusuna basiniz !?';
                 GO TO L1;
              END;
              READ  FILE(R) INTO(DUMMY_PTRS);
              T1#=0;
              DO J=1 TO #OFAREAS(I);
                 DO K=1 TO #OFSUBAREAS(I);
                    T1#=T1#+1;
                    KEY_PTRS(I,J,K)=DUMMY_PTRS(T1#);
                 END;
              END;
              FREE DUMMY_PTRS;
           END;
           MSG=CLS!!ESC!!'"'!!POS(5)!!POS(10)!!
               'Soru kitabi ayrimi varmi ???Hayir?'!!
                   ESC!!'"'!!POS(5)!!POS(11)!!
               '(Ayrim varsa?Evet?giriniz)'!!DC2ETX;
           WRITE FILE(R) FROM(MSG);
           READ  FILE(R) INTO(BTFLAG);
           BT#=1;
           BTFLAG=EXCEPT(' ',BTFLAG);
           IF BTFLAG='Evet' ! BTFLAG='EVET' ! BTFLAG='evet'
                                            THEN BT#=2;
           ALLOCATE A_KEYS;
           TITLE(ANSWER_FILE)=SYSID!!'/ANSWERKEYS';
           IF RESIDENT(ANSWER_FILE) THEN DO;
              MSG=
              CLS!!'YANIT ANAHTARI '!!SYSID!!'/ANSWERKEYS'!!
                  ' ADLI KUTUKTEN OKUNUYOR...'!!CR!!ETX;
              WRITE FILE(R) FROM(MSG);
              READ FILE(ANSWER_FILE) INTO(ANSWERKEYS_REC);
```

```
            DO I=1 TO BT#;
               DO J=1 TO #OFSESSIONS;
                  DO K=1 TO #OFAREAS(J);
                     A_KEYS(I,J,K,*)=A_KEYS_2(I,J,K,*);
                  END;
               END;
            END;
            MSG=
            'YANIT ANAHTARLARI OKUNDU. DEVAM ETMEK ICIN'!!
                '?SPCFY?TUSUNA BASINIZ'!!CR!!ETX;
            WRITE FILE(R) FROM(MSG);
            READ FILE(R) INTO(STRUCT999);
            GO TO XIT;
         END;
         DO I=1 TO #OFSESSIONS;
            SUBSTR(SCREEN217,250,5)=' :?'!!I!!' ';
            IF BT#=1 THEN SUBSTR(SCREEN217,335,11)=
                     ':?Ayrim yok';
            DO L=1 TO BT#;
               IF BT#=2 THEN IF L=1 THEN
                  SUBSTR(SCREEN217,335,3)=':?A';
                  ELSE SUBSTR(SCREEN217,335,3)=':?B';
                DO J=1 TO #OFAREAS(I);
                   SUBSTR(SCREEN217,566,2)=J;
                   ERAKEYS='';
                   DO K=1 TO #OFSUBAREAS(I);
                      SUBSTR(SCREEN217,577,1)=K;
                      SUBSTR(SCREEN217,1841,80)=' ';
                   L2:WRITE FILE(R) FROM(SCREEN217);
                      READ FILE(R)  INTO(SAKEYS);
                      IF SUBSTR(ERAKEYS,1,1)=ESC THEN DO;
                         SUBSTR(SCREEN217,1841,80)=
                   '???SPCFY yerine XMIT tusuna basiniz?';
                         GO TO L2;
                      END;
                      ERAKEYS=ERAKEYS !! SAKEYS;
                   END;
                  /* WRITE FILE(R) FROM(ERAKEYS);      */
                   DO KK=1 TO LENGTH(ERAKEYS);
                    A_KEYS(L,I,J,KK)=SUBSTR(ERAKEYS,KK,1);
                   END;
                END;
            END;
         END;
/*       DO I=1 TO #OFSESSIONS;                  */
/*          DO J=1 TO #OFAREAS(I);               */
/*             DO K=1 TO 5 ;                     */
/*             PUT DATA(I,J,A_KEYS(1,I,J,K));*/
/*             READ FILE(R) INTO(STRUCT999); */
/*             END;                              */
/*          END;                                 */
/*       END;                                    */
         DO I=1 TO BT#;
            DO J=1 TO #OFSESSIONS;
               DO K=1 TO #OFAREAS(J);
                  A_KEYS_2(I,J,K,*)=A_KEYS(I,J,K,*);
               END;
```

```
            END;
         END;
         TITLE(ANSWER_FILE)=SYSID!!'/ANSWERKEYS';
         OPEN FILE(ANSWER_FILE) OUTPUT;
         WRITE FILE(ANSWER_FILE) FROM(ANSWERKEYS_REC);
         CLOSE FILE(ANSWER_FILE) ENV(LOCK);
         MSG='ANAHTARLAR '!!SYSID!!'/ANSWERKEYS ADLI'!!
             ' KUTUGE YAZILDI DEVAM ICIN?SPCFY?A BASINIZ'!!
             CR!!ETX;
         WRITE FILE(R) FROM(MSG);
         READ FILE(R) INTO(SRTUCT999);
         XIT:
END GET_ANSWER_KEYS;

EVAL_RAW_SCORES:PROCEDURE;
  DCL COUNT#   FIXED(6) INIT(0);
  DCL TRUE BIT(1) INIT('1'B);
  DCL (B#,S#) PIC'9';
  DCL A#      PIC'99';
  DCL (I,J,K) FIXED(3);
  DCL (COR#,INCOR#,DOUBLE#,EMPTY#) PIC'999';

  ERROR_OUT:PROCEDURE(N);
      DCL N FIXED;
      PUT EDIT('ERROR:',N)(A,F(2));
      END ERROR_OUT;

      CHECK_ID:PROCEDURE(ID) RETURNS(BIT(1));
          DCL ID PIC '999999';
          DCL APP# PIC'999999';
          DCL N(6) PIC'9' DEF APP# POS(1);
           APP#=ID;
           N(6)=9-MOD(N(1)*2+N(2)*3+N(3)*4+N(4)*5+N(5)*6,9);
           RETURN(APP#=ID);
      END CHECK_ID;
           DCL NN PIC '999999';
           TITLE(EXAMOPTIC)=SYSID !! '/EXAMOPTIC';
           OPEN FILE(EXAMOPTIC) INPUT;
           ON ENDFILE(EXAMOPTIC) BEGIN;
              CLOSE FILE(EXAM) ENV(LOCK);
              CLOSE FILE(EXAMOPTIC) ENV(LOCK);
              GO TO XIT;
           END;
           TITLE(EXAM)=SYSID !! '/EXAMINATION';
           OPEN FILE(EXAM) UPDATE;
           DO WHILE(TRUE);
              READ FILE(EXAMOPTIC) INTO(EXAMOPTIC_REC);
              COUNT#=COUNT#+1;
           /* IF COUNT#=2 THEN STOP;
              WRITE FILE(R) FROM(EXAMOPTIC_REC);*/
              S#=EXAMOPTIC_REC.SESSION;
              A#   =EXAMOPTIC_REC.AREA;
              IF EXAMOPTIC_REC.BOOKLET_TYPE='A' THEN B#=1;
                                                ELSE B#=2;
              IF ¬ CHECK_ID(EXAMOPTIC_REC.APPLICATION#)
              THEN DO;
                 CALL ERROR_OUT(1);
```

```
         GO TO XITLOOP;
      END;
      READ FILE(EXAM) INTO(EXAM_REC)
           INDEX(FLOOR(
                 EXAMOPTIC_REC.APPLICATION#/10)-1);
      IF S# < 1 ! S# > #OFSESSIONS THEN DO;
         CALL ERROR_OUT(2);
         GO TO XITLOOP;
      END;
      IF A# < 1 ! A# > #OFAREAS(S#) THEN DO;
         CALL ERROR_OUT(3);
         GO TO XITLOOP;
      END;
      IF EXAM_REC.READ_FLAG(S#) ¬= 0  THEN DO;
         CALL ERROR_OUT(5);
         GO TO XITLOOP;
      END;
     /*IF EXAM_REC.AREA(S#) ¬= A# THEN DO;
         CALL ERROR_OUT(6);
         GO TO XITLOOP;
      END;                    */
      IF EXAMOPTIC_REC.ATTENDANCE = 2 THEN DO;
         EXAM_REC.SESSION.ATTENDANCE(S#)=2;
         REWRITE FILE(EXAM) FROM(EXAM_REC);
         GO XITLOOP;
      END;
   DO I=1 TO #OFSUBAREAS(S#);
      IF I=1 THEN K=1;
             ELSE K=KEY_PTRS(S#,A#,I-1);
      COR#=0;
      INCOR#=0;
      DOUBLE#=0;
      EMPTY#=0;
      DO J=K TO KEY_PTRS(S#,A#,I);

         IF A_KEYS(B#,S#,A#,J)='+' THEN
            GO TO SKIPTHIS
         IF EXAMOPTIC_REC.ANSWER(J)='*'
         THEN DOUBLE#=DOUBLE#+1;
         ELSE
         IF EXAMOPTIC_REC.ANSWER(J)=' '
         THEN EMPTY#=EMPTY#+1;
         ELSE
         IF EXAMOPTIC_REC.ANSWER(J) ¬=
           A_KEYS(B#,S#,A#,J) THEN INCOR#=INCOR#+1;
         ELSE COR#=COR#+1;
      SKIPTHIS:
      END;
      EXAM_REC.CORRECT(S#,I)=COR#;
      EXAM_REC.INCORRECT(S#,I)=INCOR#+DOUBLE#;
      EXAM_REC.RAW(S#,I)=COR#-INCOR#/4;
   END;

   REWRITE FILE(EXAM) FROM(EXAM_REC);
   XITLOOP:
   END;
XIT: WRITE FILE(R) FROM(SCREEN218);
```

```
   READ  FILE(R) INTO(STRUCT999);
END EVAL_RAW_SCORES;

EVAL_STANDARD_SCORES:PROCEDURE;
   DCL (NN,TOT,TOT2,SDS,MEAN)(4,5) FLOAT;
   WRITE FILE(R) FROM(SCREEN219);
   READ  FILE(R) INTO(STRUCT999);
   ON ENDFILE(EXAM) BEGIN;
      MSG='ORTALAMA VE STANDART SAPMALAR BULUNDU'
          !!CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
      CLOSE FILE(EXAM);
      GO TO LL;
   END;
   TITLE(EXAM)=SYSID!!'/EXAMINATION';
   OPEN FILE(EXAM) INPUT;
   DO WHILE('1'B);
      READ FILE(EXAM) INTO(EXAM_REC);
      DO I=1 TO #OFSESSIONS;
         IF EXAM_REC.AREA(I) = 0 THEN
            GO TO SKIPTHIS;
         DO J=1 TO #OFSUBAREAS(I);
          NN(I,J)=NN(I,J)+1;
          TOT(I,J)=TOT(I,J)+EXAM_REC.CORRECT(I,J);
          TOT2(I,J)=TOT2(I,J)+EXAM_REC.CORRECT(I,J)
                                          **2;
         END;
      SKIPTHIS:
      END;
   END;
LL:PUT DATA(NN);
   MSG='STANDART PUAN HESAPLANIYOR.'!!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   DO I=1 TO #OFSESSIONS;
         DO J=1 TO #OFSUBAREAS(I);
            SDS(I,J)=SQRT((TOT2(I,J)-(TOT(I,J)
                  **2/NN(I,J)))/(NN(I,J)-1));
            MEAN(I,J)=TOT(I,J)/NN(I,J);
         END;
   END;
   ON ENDFILE(EXAM) BEGIN;
      CLOSE FILE(EXAM) ENV(LOCK);
      MSG='STANDARD PUANLAR HESAPLANDI'!!
          CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
      MSG='GERI DONMEK ICIN?SPCFY?TUSUNA BASINIZ'
          !!CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
      READ FILE(R) INTO(STRUCT999);
      GO TO XIT;
   END;
   OPEN FILE(EXAM) UPDATE;
   DO WHILE('1'B);
      READ FILE(EXAM) INTO(EXAM_REC);
      DO I=1 TO #OFSESSIONS;
         IF EXAM_REC.AREA(I) = 0 THEN
            GO TO SKIPTHIS2;
```

```
                  DO J=1 TO #OFSUBAREAS(I);
                     EXAM_REC.STANDARD(I,J)=
                      10*((EXAM_REC.CORRECT(I,J)-MEAN(I,J))
                      /SDS(I,J))+50;
                   EXAM_REC.WEIGHTED(I)=EXAM_REC.WEIGHTED(I)+
                      EXAM_REC.STANDARD(I,J);
                  END;
               SKIPTHIS2:
               END;
               REWRITE FILE(EXAM) FROM(EXAM_REC);
            END;
         XIT:
END EVAL_STANDARD_SCORES;

EVAL_WEIGHTED_SCORES:PROCEDURE;
            WRITE FILE(R) FROM(SCREEN220);
            READ  FILE(R) INTO(STRUCT999);
            WRITE FILE(R) FROM(SCREEN221);
            READ  FILE(R) INTO(STRUCT221);
END EVAL_WEIGHTED_SCORES;

TRANS_EXAM_DATA:PROCEDURE;
   DCL TRUE BIT(1) INIT('1'B);
   DCL (I,J,K) FIXED;
   TITLE(CANDIDATE)=SYSID !! '/CANDIDATE';
   TITLE(EXAM)=     SYSID !! '/EXAMINATION';
   OPEN FILE(CANDIDATE) UPDATE;
   OPEN FILE(EXAM) INPUT;
   ON ENDFILE(EXAM) BEGIN;
      MSG='Bilgiler Aday kutugune aktarildi.' !!CR!!ETX;
      WRITE FILE(R) FROM(MSG);
      CLOSE FILE(CANDIDATE) ENV(LOCK);
      CLOSE FILE(EXAM);
      MSG='DEVAM ETMEK ICIN?SPCFY?TUSUNA BASINIZ.'
          !!CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
      READ FILE(R) INTO(STRUCT999);
      GO TO XIT;
   END;
   MSG='Degerlendirme bilgileri Aday kutugune aktariliyor'
       !!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
/* ON ERROR BEGIN;
      PUT DATA(CANDIDATE_REC.SESSION,EXAM_REC.SESSION);
      STOP;
   END;          */
   DO WHILE(TRUE);
      READ FILE(EXAM) INTO(EXAM_REC);
      READ FILE(CANDIDATE) INTO(CANDIDATE_REC)
           INDEX(FLOOR(EXAM_REC.APPLICATION#/10)-1);
     DO I=1 TO #OFSESSIONS;
      CANDIDATE_REC.WEIGHTED(I)=EXAM_REC.WEIGHTED(I);
      DO J=1 TO #OFSUBAREAS(I);
       CANDIDATE_REC.CORRECT(I,J)=EXAM_REC.CORRECT(I,J);
       CANDIDATE_REC.INCORRECT(I,J)=EXAM_REC.INCORRECT(I,J);
       CANDIDATE_REC.RAW(I,J)=EXAM_REC.RAW(I,J);
       CANDIDATE_REC.STANDARD(I,J)=EXAM_REC.STANDARD(I,J);
```

```
      END;
     END;
     REWRITE FILE(CANDIDATE) FROM(CANDIDATE_REC);
   END;
   XIT:
END TRANS_EXAM_DATA;

        CREATE_SPEC_FILE:PROCEDURE;
        END CREATE_SPEC_FILE;
DCL SPCFYOKFLAG          BIT(1) INIT('1'B);
DCL TYPE                 FIXED;
DO WHILE(TYPE ¬= 99);
DO WHILE(SPCFYOKFLAG);
   SPCFYOKFLAG='0'B;
   WRITE FILE(R) FROM(SCREEN215);
   READ  FILE(R) INTO(STRUCT999);
   IF SPCFY(1) = ESC THEN DO;
      IF ( SPCFY(3) = POS(15) ! SPCFY(3) = POS(79))&
         ( SPCFY(4) = POS(6)  ! SPCFY(4) = POS(8)   !
           SPCFY(4) = POS(10) ! SPCFY(4) = POS(12)  !
            SPCFY(4) = POS(14) ! SPCFY(4) = POS(22) )
      THEN DO;
           TYPE=INDEX(ASCII,SPCFY(4));
           TYPE=TYPE/2-2;
           IF SPCFY(3) = POS(79) THEN TYPE=99;
      END;
      ELSE TYPE=0;
      IF TYPE = 0 THEN DO;
         SUBSTR(SCREEN215,1842,80)=BLINK!!
               'Gecersiz konumda SPCFY tusuna bastiniz.';
         SPCFYOKFLAG='1'B;
      END;
   END;
   ELSE DO;
      SUBSTR(SCREEN215,1842,80)=BLINK!!
            'XMIT yerine SPCFY tusunu kullaniniz.';
      SPCFYOKFLAG='1'B;
   END;
END;
SUBSTR(SCREEN215,1842,80)=' ';
IF TYPE ¬= 99 THEN
   IF TYPE = 1 THEN CALL GET_ANSWER_KEYS;ELSE
   IF TYPE = 2 THEN CALL EVAL_RAW_SCORES; ELSE
   IF TYPE = 3 THEN CALL EVAL_STANDARD_SCORES; ELSE
   IF TYPE = 4 THEN CALL EVAL_WEIGHTED_SCORES; ELSE
   IF TYPE = 5 THEN CALL TRANS_EXAM_DATA;
SPCFYOKFLAG='1'B;
END;
END EVAL_OF_EXAM;

PRINT_OUT:PROCEDURE;
 PRINT_ANSWER_SHEETS:PROCEDURE;
    DCL S#      PIC'9';
    ON ENDPAGE(SYSPRINT);
    TITLE(CANDIDATE)=SYSID!!'/CANDIDATE';
    OPEN FILE(CANDIDATE) INPUT;
    MSG='CEVAP KAGITLARI DOKULUYOR'!!CR!!LF!!ETX;
```

```
  WRITE FILE(R) FROM(MSG);
  ON ENDFILE(TEMP) BEGIN;
     CLOSE FILE(TEMP);
     CLOSE FILE(SYSPRINT);
     GO TO XIT;
  END;
  DO S#=1 TO #OFSESSIONS;
     TITLE(TEMP)=SYSID!!'/TEMP/'!!S#;
     SORT TEMP_REC ON ASCENDING KEY
          (TEMP_REC.CENTER,
           TEMP_REC.OUTGOING#);
     MSG=S#!!'.OTURUM'!!CR!!LF!!ETX;
     WRITE FILE(R) FROM(MSG);
     OPEN FILE(TEMP) INPUT;
     OPEN FILE(SYSPRINT);
     DO WHILE('1'B);
        READ FILE(TEMP) INTO(TEMP_REC);
        READ FILE(CANDIDATE) INTO(CANDIDATE_REC)
             INDEX(FLOOR(TEMP_REC.APPLICATION#/10)-1);
        PUT SKIP(7) EDIT (SYSID,' SINAV',
            'OTURUM:',S#,
            'SINAVA GIRECEGI ALAN:',TEMP_REC.AREA,
             TEMP_REC.APPLICATION#,
             CANDIDATE_REC.SURNAME,
             CANDIDATE_REC.NAME,
             TEMP_REC.ROOM#,
             CANDIDATE_REC.SEAT#(S#),
             TEMP_REC.CENTER,'-',TEMP_REC.BOX#,
                                 TEMP_REC.BOXSEQ#,
             TEMP_REC.CENTER,'-',TEMP_REC.OUTGOING#)
            (COL(35),A(5),A,COL(65),A,P'9',SKIP(15),
             COL(44),A,P'Z9',
             SKIP(13),COL(16),P'999999',
             SKIP( 3),COL(16),A,
             SKIP( 3),COL(16),A,
             SKIP(28),COL(13),P'99999',COL(26),P'999',
             SKIP(3),COL(12),P'99',A,P'99',P'999',
                     COL(24),P'99',A,P'99999');
     END;
  XIT:END;
  MSG='CEVAP KAGITLARI DOKULDU'!!CR!!LF!!ETX;
  WRITE FILE(R) FROM(MSG);
  MSG='GERI DONUS ICIN?SPCFY?TUSUNA BASINIZ'!!CR!!LF!!ETX;
  WRITE FILE(R) FROM(MSG);
  READ FILE(R) INTO(STRUCT999);
END PRINT_ANSWER_SHEETS;
PRINT_ROOM_ID_CARDS:PROCEDURE;
  DCL S#      PIC'9';
  DCL PROCTOR# PIC'9';
  ON ENDPAGE(SYSPRINT);
  TITLE(BUILDING)=SYSID!!'/BUILDING';
  OPEN FILE(BUILDING) INPUT;
  MSG='SALON TANITIM KARTLARI DOKULUYOR'!!CR!!LF!!ETX;
  WRITE FILE(R) FROM(MSG);
  ON ENDFILE(ROOM) BEGIN;
     CLOSE FILE(ROOM);
     CLOSE FILE(SYSPRINT);
     GO TO XIT;
```

```
   END;
   DO S#=1 TO #OFSESSIONS;
      TITLE(ROOM)=SYSID!!'/ROOM/'!!S#;
      SORT ROOM_REC ON ASCENDING KEY
           (ROOM_REC.CENTER,
            ROOM_REC.BUILDING#,
            ROOM_REC.ROOM#);
      MSG=S#!!'.OTURUM'!!CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
      OPEN FILE(ROOM) INPUT;
      OPEN FILE(SYSPRINT);
      DO WHILE('1'B);
         READ FILE(ROOM) INTO(ROOM_REC);
         PROCTOR#=CEIL(ROOM_REC.#OFASSIGNED/25);
         READ FILE(BUILDING) INTO(BUILDING_REC)
              INDEX(ROOM_REC.BUILDING@);
         PUT SKIP(15) EDIT (SYSID,' SINAV','OTURUM:',S#,
                  'SALON TANITIM KARTI',
                  ROOM_REC.ROOM#,
                  ROOM_REC.LINE1,ROOM_REC.LINE2,
                  BUILDING_REC.BUILDING#,
                  BUILDING_REC.LINE1,BUILDING_REC.LINE2,
                  ROOM_REC.#OFASSIGNED,
                  PROCTOR#)
             (COL(30),A(5),A,COL(55),A,P'9',
              SKIP(2),COL(26),A,
              SKIP(8),COL(22),P'99999',
              SKIP(9),COL(22),A,SKIP(1),COL(22),A,
              SKIP(6),COL(22),P'99999',SKIP(1),COL(22),A,
              SKIP,COL(22),A,
              SKIP(4),COL(23),P'999',
              SKIP(1),COL(40),P'9');
      END;
   XIT:END;
   MSG='SALON TANITIM KARTLARI DOKULDU'!!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   MSG='GERI DONUS ICIN?SPCFY?TUSUNA BASINIZ'!!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   READ FILE(R) INTO(STRUCT999);
END PRINT_ROOM_ID_CARDS;

PRINT_ROOM_EXAM_REPORTS:PROCEDURE;
   DCL S#      PIC'9';
   ON ENDPAGE(SYSPRINT);
   TITLE(BUILDING)=SYSID!!'/BUILDING';
   OPEN FILE(BUILDING) INPUT;
   MSG='SALON SINAV TUTANAKLARI DOKULUYOR'!!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   ON ENDFILE(ROOM) BEGIN;
      CLOSE FILE(ROOM);
      CLOSE FILE(SYSPRINT);
      GO TO XIT;
   END;
   DO S#=1 TO #OFSESSIONS;
      TITLE(ROOM)=SYSID!!'/ROOM/'!!S#;
      SORT ROOM_REC ON ASCENDING KEY
           (ROOM_REC.CENTER,
```

```
            ROOM_REC.BUILDING#,
            ROOM_REC.ROOM#);
      MSG=S#!!'.OTURUM'!!CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
      OPEN FILE(ROOM) INPUT;
      OPEN FILE(SYSPRINT);
      DO WHILE('1'B);
         READ FILE(ROOM) INTO(ROOM_REC);
         READ FILE(BUILDING) INTO(BUILDING_REC)
              INDEX(ROOM_REC.BUILDING@);
         PUT SKIP(57) EDIT ('OTURUM:',S#,
             SYSID,' SINAV',
             '(',EXAMDATE(S#),' ',EXAMHOUR(S#),')',
             ROOM_REC.ROOM#,
             ILLER(ROOM_REC.CENTER),ROOM_REC.BUILDING#,
             ROOM_REC.#OFASSIGNED,BUILDING_REC.LINE1,
             ROOM_REC.#OFASSIGNED,BUILDING_REC.LINE2,
             ROOM_REC.LINE1,ROOM_REC.LINE2)
            (         COL(34),A,P'9',
             SKIP(1),COL(24),A,A,
             SKIP(1),COL(30),A,A,A,A,A,COL(69),P'99999',
             SKIP(2),COL(46),A,COL(69),P'99999',
             SKIP(2),COL(11),P'999',COL(46),A,
             SKIP(1),COL(11),P'999',COL(46),A,
             SKIP(1),COL(46),A,
             SKIP(1),COL(46),A);
      END;
   XIT:END;
   MSG='SALON SINAV TUTANAKLARI DOKULDU'!!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   MSG='GERI DONUS ICIN?SPCFY?TUSUNA BASINIZ'!!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   READ FILE(R) INTO(STRUCT999);
END PRINT_ROOM_EXAM_REPORTS;

PRINT_EXTRA_EXAM_DOC_UTIL_SCHEDULE:PROCEDURE;
   DCL S#       PIC'9';
   DCL OLDBUILD# PIC'99999' INIT(0);
   ON ENDPAGE(SYSPRINT);
   TITLE(BUILDING)=SYSID!!'/BUILDING';
   OPEN FILE(BUILDING) INPUT;
   MSG='BINA YEDEK SINAV EVRAKI KUL CIZ. DOKULUYOR'
       !!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   ON ENDFILE(ROOM) BEGIN;
      CLOSE FILE(ROOM);
      CLOSE FILE(SYSPRINT);
      GO TO XIT;
   END;
   DO S#=1 TO #OFSESSIONS;
      TITLE(ROOM)=SYSID!!'/ROOM/'!!S#;
      SORT ROOM_REC ON ASCENDING KEY
           (ROOM_REC.CENTER,
            ROOM_REC.BUILDING#,
            ROOM_REC.ROOM#);
      MSG=S#!!'.OTURUM'!!CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
```

```
      OPEN FILE(ROOM) INPUT;
      OPEN FILE(SYSPRINT);
      DO WHILE('1'B);
         RA: READ FILE(ROOM) INTO(ROOM_REC);
         IF ROOM_REC.BUILDING# = OLDBUILD# THEN GO TO RA;
         OLDBUILD#=ROOM_REC.BUILDING#;
         READ FILE(BUILDING) INTO(BUILDING_REC)
              INDEX(ROOM_REC.BUILDING@);
         PUT SKIP(43) EDIT (
             SYSID,' SINAV','OTURUM:',S#,
             'ÇLÇ.....:',ILLER(ROOM_REC.CENTER),
             '(',EXAMDATE(S#),' ',EXAMHOUR(S#),')',
             'BÇNA NO.:',BUILDING_REC.BUILDING#,
             'BÇNA ADI:',BUILDING_REC.LINE1,
             BUILDING_REC.LINE2)
            (         COL(20),A,A,COL(42),A,P'9',
             SKIP(1),COL(60),A,A,
             SKIP(1),COL(18),A,A,A,A,A,COL(60),A,P'99999',
             SKIP(2),COL(60),A,A,
             SKIP(1),COL(69),A);
      END;
   XIT:END;
   MSG='BINA YEDEK SINAV EVRAKI KUL. CIZ. DOKULDU'
       !!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   MSG='GERI DONUS ICIN?SPCFY?TUSUNA BASINIZ'
       !!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   READ FILE(R) INTO(STRUCT999);
END PRINT_EXTRA_EXAM_DOC_UTIL_SCHEDULE;

PRINT_ROOM_CANDIDATE_ATTENDANCE_LIST:PROCEDURE;
   DCL S#      PIC'9';
   DCL I#      PIC'999';
   ON ENDPAGE(SYSPRINT);
   TITLE(CANDIDATE)=SYSID!!'/CANDIDATE';
   TITLE(BUILDING)=SYSID!!'/BUILDING';
   OPEN FILE(CANDIDATE) INPUT;
   OPEN FILE(BUILDING)  INPUT;
   MSG='SALON ADAY YOKLAMA LISTELERI DOKULUYOR'
       !!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   ON ENDFILE(ROOM) BEGIN;
      CLOSE FILE(TEMP);
      CLOSE FILE(ROOM);
      CLOSE FILE(SYSPRINT);
      GO TO XIT;
   END;
   DO S#=1 TO #OFSESSIONS;
      TITLE(TEMP)=SYSID!!'/TEMP/'!!S#;
      SORT TEMP_REC ON ASCENDING KEY
           (TEMP_REC.CENTER,
            TEMP_REC.BUILDING#,
            TEMP_REC.ROOM#,
            TEMP_REC.APPLICATION#);
      TITLE(ROOM)=SYSID!!'/ROOM/'!!S#;
      SORT ROOM_REC ON ASCENDING KEY
```

```
          (ROOM_REC.CENTER,
           ROOM_REC.BUILDING#,
           ROOM_REC.ROOM#);
      MSG=S#!!'.OTURUM'!!CR!!LF!!ETX;
      WRITE FILE(R) FROM(MSG);
      OPEN FILE(TEMP) INPUT;
      OPEN FILE(ROOM) INPUT;
      OPEN FILE(SYSPRINT) ENV(PAGESIZE=72);
      DO WHILE('1'B);
         READ FILE(ROOM) INTO(ROOM_REC);
         READ FILE(BUILDING) INTO(BUILDING_REC)
              INDEX(ROOM_REC.BUILDING@);
         PUT PAGE    EDIT (SYSID,' SINAV','OTURUM:',S#,
             ROOM_REC.ROOM#,
             '(',EXAMDATE(S#),' ',EXAMHOUR(S#),')',
             '1',
             ILLER(ROOM_REC.CENTER),BUILDING_REC.LINE1,
                                    BUILDING_REC.LINE2,
             BUILDING_REC.BUILDING#,ROOM_REC.LINE1,
                                    ROOM_REC.LINE2)
             (COL(33),A,A,COL(56),A,P'9',
              SKIP(1),COL(69),P'99999',
              SKIP(1),COL(30),A,A,A,A,A,
              SKIP(1),COL(72),A,
              SKIP(7),COL(15),A,COL(43),A,
              SKIP(1),COL(43),A,
              SKIP(2),COL(15),P'99999',COL(43),A,
              SKIP(1),COL(43),A);
         PUT SKIP(2);
         DO I#=1 TO ROOM_REC.#OFASSIGNED;
              READ FILE(TEMP) INTO(TEMP_REC);
              READ FILE(CANDIDATE) INTO(CANDIDATE_REC)
                 INDEX(FLOOR(TEMP_REC.APPLICATION#/10)-1);
              PUT SKIP(1) EDIT(CANDIDATE_REC.SEAT#(S#),
                              CANDIDATE_REC.SURNAME,
                               CANDIDATE_REC.NAME,
                               TEMP_REC.APPLICATION#)
                  (COL(21),P'999',COL(27),A,A,COL(66),
                   P'999999');
         END;
         PUT LINE(66) EDIT(ROOM_REC.#OFASSIGNED,
                           ROOM_REC.#OFASSIGNED)
                          (COL(21),P'999',COL(31),P'999');
      END;
   XIT:END;
   MSG='SALON ADAY YOKLAMA LISTELERI DOKULDU'!!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   MSG='GERI DONUS ICIN?SPCFY?TUSUNA BASINIZ'!!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   READ FILE(R) INTO(STRUCT999);
END PRINT_ROOM_CANDIDATE_ATTENDANCE_LIST;

PRINT_BUILDING_ROOM_LIST:PROCEDURE;
   DCL S#       PIC'9';
   DCL OLDBUILD# PIC'99999' INIT(0);
   ON ENDPAGE(SYSPRINT);
   TITLE(BUILDING)=SYSID!!'/BUILDING';
```

```
OPEN FILE(BUILDING) INPUT;
MSG='BINA YEDEK SINAV EVRAKI KUL CIZ. DOKULUYOR'
    !!CR!!LF!!ETX;
WRITE FILE(R) FROM(MSG);
ON ENDFILE(ROOM) BEGIN;
   CLOSE FILE(ROOM);
   CLOSE FILE(SYSPRINT);
   GO TO XIT;
END;
DO S#=1 TO #OFSESSIONS;
   TITLE(ROOM)=SYSID!!'/ROOM/'!!S#;
   SORT ROOM_REC ON ASCENDING KEY
        (ROOM_REC.CENTER,
         ROOM_REC.BUILDING#,
         ROOM_REC.ROOM#);
   MSG=S#!!'.OTURUM'!!CR!!LF!!ETX;
   WRITE FILE(R) FROM(MSG);
   OPEN FILE(ROOM) INPUT;
   OPEN FILE(SYSPRINT) ENV(PAGESIZE=72);
   DO WHILE('1'B);
      RA: READ FILE(ROOM) INTO(ROOM_REC);
      IF ROOM_REC.BUILDING# ¬= OLDBUILD# THEN DO;
         OLDBUILD#=ROOM_REC.BUILDING#;
         READ FILE(BUILDING) INTO(BUILDING_REC)
              INDEX(ROOM_REC.BUILDING@);
         PUT PAGE EDIT(SYSID,' SINAV','OTURUM:',S#,
             '(',EXAMDATE(S#),' ',EXAMHOUR(S#),')',
             'BİNADAKİ SALONLAR LİSTESİ',
             'BİNA NO  :',ROOM_REC.BUILDING#,
             'BİNA ADI :',BUILDING_REC.LINE1,
                          BUILDING_REC.LINE2,
     'SALON NO  ALANI  KAPASİTESİ  ATANAN ADAY SAYISI',
             'SALONUN A@IK ADI',REPEAT('=',100))
             (COL(32),A,A,COL(46),A,P'9',
             COL(56),A,A,A,A,A,
             SKIP(2),COL(37),A,
             SKIP(2),COL(3),A,P'99999',
             SKIP(2),COL(3),A,A,
             SKIP(1),COL(13),A,
             SKIP(2),COL(3),A,COL(72),A,
             SKIP(1),COL(3),A);
      END;
      PUT SKIP EDIT (
                          ROOM_REC.ROOM#,ROOM_REC.AREA,
                          ROOM_REC.ROOM_CAPACITY,
                          ROOM_REC.#OFASSIGNED,
                          ROOM_REC.LINE1,
                          ROOM_REC.LINE2)
         (COL(4),P'99999',
          COL(14),P'99',
          COL(23),P'999',
          COL(39),P'999',
          COL(58),A,X(1),A);
   END;
XIT:END;
MSG='BINADAKI SALONLAR LISTESI DOKULDU'!!CR!!LF!!ETX;
WRITE FILE(R) FROM(MSG);
```

```
    MSG='GERI DONUS ICIN?SPCFY?TUSUNA BASINIZ'!!CR!!LF!!ETX;
    WRITE FILE(R) FROM(MSG);
    READ FILE(R) INTO(STRUCT999);
  END PRINT_BUILDING_ROOM_LIST;

DCL SPCFYOKFLAG          BIT(1) INIT('1'B);
DCL TYPE                 FIXED;
DO WHILE(TYPE ¬= 99);
DO WHILE(SPCFYOKFLAG);
   SPCFYOKFLAG='0'B;
   WRITE FILE(R) FROM(SCREEN222);
   READ  FILE(R) INTO(STRUCT999);
   IF SPCFY(1) = ESC THEN DO;
      IF ( SPCFY(3) = POS(15) ! SPCFY(3) = POS(79))&
         ( SPCFY(4) = POS(6)  ! SPCFY(4) = POS(7)   !
           SPCFY(4) = POS(8)  ! SPCFY(4) = POS(9)   !
           SPCFY(4) = POS(10) ! SPCFY(4) = POS(11)  !
           SPCFY(4) = POS(12) ! SPCFY(4) = POS(13)  !
           SPCFY(4) = POS(14) ! SPCFY(4) = POS(22)  )
      THEN DO;
           TYPE=INDEX(ASCII,SPCFY(4));
           TYPE=TYPE-5;
           IF SPCFY(3) = POS(79) THEN TYPE=99;
      END;
      ELSE TYPE=0;
      IF TYPE = 0 THEN DO;
         SUBSTR(SCREEN222,1842,80)=BLINK!!
               'Gecersiz konumda SPCFY tusuna bastiniz.';
         SPCFYOKFLAG='1'B;
      END;
   END;
   ELSE DO;
      SUBSTR(SCREEN222,1842,80)=BLINK!!
            'XMIT yerine SPCFY tusunu kullaniniz.';
      SPCFYOKFLAG='1'B;
   END;
END;
SUBSTR(SCREEN222,1842,80)=' ';
IF TYPE ¬= 99 THEN
   IF TYPE=1 THEN CALL PRINT_ANSWER_SHEETS;ELSE
   IF TYPE=3 THEN CALL PRINT_ROOM_ID_CARDS;ELSE
   IF TYPE=4 THEN CALL PRINT_ROOM_CANDIDATE_ATTENDANCE_LIST;
   ELSE
   IF TYPE=5 THEN CALL PRINT_ROOM_EXAM_REPORTS;ELSE
   IF TYPE=7 THEN CALL PRINT_EXTRA_EXAM_DOC_UTIL_SCHEDULE;
   ELSE
   IF TYPE=8 THEN CALL PRINT_BUILDING_ROOM_LIST;
SPCFYOKFLAG='1'B;
END;
END PRINT_OUT;

/*%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/
/*                                                        */
/*            M A I N   P R O G R A M                     */
/*                                                        */
/*%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/
ON UNDEFINEDFILE(P);
```

```
ON RECORD(R);
MYSTA =-STATION(MYSELF);
CALL GETUSERPARAMS(MYSTA);

/*%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/
/*          ALLOCATION OF DATA STRUCTURES                                 */
/*%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#%#*/

ALLOCATE CANDIDATE_REC;
ALLOCATE SUMTABLE;
ALLOCATE ATABLE;
ALLOCATE STRUCT213;
ALLOCATE A_KEYS;
ALLOCATE KEY_PTRS;

DO WHILE(TYPE ¬= 99);
DO WHILE(SPCFYOKFLAG);
   SPCFYOKFLAG='0'B;
   WRITE FILE(R) FROM(SCREEN210);
   READ  FILE(R) INTO(STRUCT999);
   IF SPCFY(1) = ESC THEN DO;
      IF ( SPCFY(3) = POS(15) ! SPCFY(3) = POS(79))&
         ( SPCFY(4) = POS(6)  ! SPCFY(4) = POS(8)   !
           SPCFY(4) = POS(10) ! SPCFY(4) = POS(22) )
      THEN DO;
           TYPE=INDEX(ASCII,SPCFY(4));
           TYPE=TYPE/2-2;
           IF SPCFY(3) = POS(79) THEN TYPE=99;
      END;
      ELSE TYPE=0;
      IF TYPE = 0 THEN DO;
         SUBSTR(SCREEN210,1842,80)=BLINK!!
               'Gecersiz konumda SPCFY tusuna bastiniz.';
         SPCFYOKFLAG='1'B;
      END;
   END;
   ELSE DO;
      SUBSTR(SCREEN210,1842,80)=BLINK!!
            'XMIT yerine SPCFY tusunu kullaniniz.';
      SPCFYOKFLAG='1'B;
   END;
END;
SUBSTR(SCREEN210,1842,80)=' ';
IF TYPE ¬= 99 THEN CALL STARTSYS(TYPE);
SPCFYOKFLAG='1'B;
END;
  MSG=CLS!!ESC!!'"2-'!!BLINK!!
      'Geri donus icin lutfen bekleyiniz'!!
       ESC!!'"o7'!!DC2ETX;
  WRITE FILE(R) FROM(MSG);
  CLOSE FILE(R);
END EXAMINATION;
```

```
%-------------------------------------------------------------%
%                                                             %
%                   REX/LIBRARY/STARTJOB                      %
%                                                             %
%-------------------------------------------------------------%

BEGIN
  BOOLEAN PROCEDURE STARTJOB(S); STRING S;
  BEGIN
    FILE D(KIND=DISK);
    FILE Z(KIND=DISK,MAXRECSIZE=15,BLOCKSIZE=420);
    ARRAY WAITFILE [0:1],INPUTFILE [0:5];
    STRING MYSTA;
    POINTER P,PP;
    MYSTA:=STRING(-MYSELF.STATION,*);
    REPLACE P:=POINTER(WAITFILE,8)
            BY "0" FOR 4-LENGTH(MYSTA),MYSTA,"/WAIT2.";
    REPLACE D.TITLE BY P;
    IF D.RESIDENT THEN REMOVEFILE(P);
    REPLACE PP:=POINTER(INPUTFILE,8) BY S,".";
    REPLACE Z.TITLE BY PP;
    ZIP WITH Z;
    WHILE NOT D.RESIDENT DO; REMOVEFILE(P);
  END;
  EXPORT STARTJOB;
  FREEZE(TEMPORARY);
END.
```

```
%-------------------------------------------------------------%
%                                                             %
%                   REX/LIBRARY/SUBSYSTEM                     %
%                                                             %
%-------------------------------------------------------------%
BEGIN
  BOOLEAN PROCEDURE SUBSYSTEM(I); INTEGER I;
  BEGIN
    FILE D(KIND=DISK);
    ARRAY WAITFILE [0:1],JOBARRAY [0:99];
    STRING MYSTA;
    POINTER P;
    MYSTA:=STRING(-MYSELF.STATION,*);
    REPLACE P:=POINTER(WAITFILE,8)
             BY "0" FOR 4-LENGTH(MYSTA),MYSTA,"/WAIT1.";
    REPLACE D.TITLE BY P;
    IF D.RESIDENT THEN REMOVEFILE(P);
    CASE I OF BEGIN
       1:REPLACE POINTER(JOBARRAY) BY
         "?BEGIN JOB;USER=REX/REX",
         ";STATION=",MYSTA,
         ";FILE D(DISK,TITLE=",P FOR 10,")",
         ";RUN OBJECT/REX/REGISTRATION",
         ";D(MYUSE=OUT);OPEN(D);LOCK(D)",
         ";END JOB";
       2:REPLACE POINTER(JOBARRAY) BY
         "?BEGIN JOB;USER=REX/REX",
         ";STATION=",MYSTA,
         ";FILE D(DISK,TITLE=",P FOR 10,")",
         ";RUN OBJECT/REX/EXAMINATION",
         ";D(MYUSE=OUT);OPEN(D);LOCK(D)",
         ";END JOB";
    ELSE:REPLACE POINTER(JOBARRAY) BY
         "?BEGIN JOB;USER=REX/REX",
         ";STATION=",MYSTA,
         ";FILE D(DISK,TITLE=",P FOR 10,")",
         ";RUN OBJECT/REX/MAINTENANCE",
         ";D(MYUSE=OUT);OPEN(D);LOCK(D)",
         ";END JOB";
    END;
    ZIP WITH JOBARRAY[*];
    WHILE NOT D.RESIDENT DO; REMOVEFILE(P);
  END;
  EXPORT SUBSYSTEM;
  FREEZE(TEMPORARY);
END.
```

```
%-------------------------------------------------------------%
%                                                             %
%                   REX/LIBRARY/FILEMANAGER                   %
%                                                             %
%-------------------------------------------------------------%

$SET INSTALLATION 1
BEGIN

  BOOLEAN PROCEDURE COPY(FILE1,FILE2);STRING FILE1,FILE2;
  BEGIN
    FILE F(KIND=DISK);
    ARRAY ZIPARRAY[0:99],FILEARRAY[0:20];
    STRING S;
    POINTER P;
    P:=POINTER(FILEARRAY);
    REPLACE P BY FILE1,".";
    REPLACE F.TITLE BY P;
    IF NOT F.RESIDENT THEN COPY:=FALSE
    ELSE BEGIN
      P:=POINTER(FILEARRAY);
      REPLACE P BY FILE2,".";
      REPLACE F.TITLE BY P;
      IF F.RESIDENT THEN COPY:=NOT REMOVEFILE(FILEARRAY);
      S:="?BEGIN JOB REX/COPY;USER=REX/REX;" CAT
         "COPY " CAT FILE1 CAT " AS " CAT FILE2 CAT
         ";END JOB";
      REPLACE POINTER(ZIPARRAY) BY S;
      ZIP WITH ZIPARRAY;
      REPLACE P BY FILE2,".";
      REPLACE F.TITLE BY P;
      WHILE NOT F.PRESENT DO;
      COPY:=TRUE;
    END;
  END;

  BOOLEAN PROCEDURE REMOVE(FILE1);STRING FILE1;
  BEGIN
    ARRAY REMOVEARRAY[0:20];
    REPLACE POINTER(REMOVEARRAY,8) BY FILE1,".";
    REMOVE:=NOT REMOVEFILE(REMOVEARRAY);
  END;

  BOOLEAN PROCEDURE CHANGE(FILE1,FILE2);STRING FILE1,FILE2;
  BEGIN
    ARRAY OLD,NEW[0:20];
    REPLACE POINTER(OLD,8) BY FILE1,".";
    REPLACE POINTER(NEW,8) BY FILE2,".";
    CHANGE:=NOT CHANGEFILE(OLD,NEW);
  END;

BOOLEAN PROCEDURE SEARCH(COMMAND,FILENAME);
STRING COMMAND,FILENAME;
BEGIN

BOOLEAN PROCEDURE DSPTOSTN(PIN,POUT,SF);
   POINTER PIN,POUT;
```

```
BOOLEAN SF;
BEGIN
   TRUTHSET T (" " OR "."),
            T1(" " OR "." OR ")");
   ARRAY    A[0:2];
   INTEGER  CNT;
   POINTER  P,P1,P2;
   DEFINE
      SCANP    =SCAN P:P WHILE = " "#,
      CASEP    =CASE FIRST(STRING8(P,1)) OF BEGIN#,
      ISCHAR(C)=(INTEGER(C))#,
      HEXF(V,I)=A[I].[47:4]:=ENTIER((V)/16);
                A[I].[43:4]:=ENTIER((V) MOD 16)#,
   DEFINEND=#;
   SF:=FALSE;
   DSPTOSTN:=FALSE;
   P:=PIN;
   SCANP;
   CASEP
      ISCHAR("="):BEGIN
                  P:=P+1;SCANP;
                  CASEP
                     ISCHAR("."):BEGIN
                                 REPLACE POUT
                                      BY 4"030000";
                                 END;
                     ISCHAR("0"):
                        BEGIN
                           P:=P+2;SCANP;
                           SCAN P1:P FOR CNT:50
                                    UNTIL IN T;
                           HEXF((54-CNT),0);
                           HEXF((50-CNT),1);
                           REPLACE POUT BY
                              POINTER(A[0],8) FOR 1,
                              4"0401" FOR 2,
                              POINTER(A[1],8) FOR 1,
                              P:P UNTIL IN T;
                        END;
                     ELSE:DSPTOSTN:=TRUE;
                     END;
                  END;
      ISCHAR("*"):BEGIN
                  P:=P+1;
                  IF P NEQ " " AND P NEQ "."
                  THEN DSPTOSTN:=TRUE;
                  ELSE BEGIN
                     SCANP;
                     SF:=TRUE;
                     CASEP
                        ISCHAR("."):BEGIN
                                    REPLACE POUT
                                         BY 4"030200";
                                    END;
                        ISCHAR("0"):
                           BEGIN
                              P:=P+2;SCANP;
```

```
                        SCAN P1:P FOR CNT:50
                                  UNTIL IN T;
                        HEXF((54-CNT),0);
                        HEXF((50-CNT),1);
                        REPLACE POUT BY
                           POINTER(A[0],8) FOR 1,
                           4"0601" FOR 2,
                           POINTER(A[1],8) FOR 1,
                           P:P UNTIL IN T;
                     END;
                  ELSE:DSPTOSTN:=TRUE;
               END;
            END;
            END;
ISCHAR("("):BEGIN
            P1:=P+1;
            SCAN P:P1 FOR CNT:50 UNTIL IN T1;
            IF P = ")" THEN BEGIN
               P:=P+1;
               IF P NEQ " " AND P NEQ "." THEN
                  DSPTOSTN:=TRUE
               ELSE BEGIN
                  SCANP;
                  CASEP
                     ISCHAR("."):
                        BEGIN
                           HEXF((54-CNT),0);
                           HEXF((50-CNT),1);
                           REPLACE POUT BY
                              POINTER(A[0],8) FOR 1
                             ,4"0301" FOR 2
                             ,POINTER(A[1],8) FOR 1
                             ,P1:P1 UNTIL IN T1;
                        END;
                     ISCHAR("0"):
                        BEGIN
                           P:=P+2;SCANP;
                           HEXF((50-CNT),0);
                           SCAN P2:P FOR CNT:50
                                     UNTIL IN T;
                           HEXF((50-CNT),1);
                           HEXF((A[0].[47:8]+
                                 A[1].[47:8]+5),2);
                           REPLACE POUT BY
                              POINTER(A[2],8) FOR 1
                             ,4"0702" FOR 2
                             ,POINTER(A[0],8) FOR 1
                             ,P1:P1 UNTIL IN T1
                             ,POINTER(A[1],8) FOR 1
                             ,P:P UNTIL IN T;
                        END;
                     ELSE:DSPTOSTN:=TRUE;
                  END;
               END;
            END ELSE DSPTOSTN:=TRUE;
            END;
ELSE:DSPTOSTN:=TRUE;
```

```
      END;
   END OF DSPTOSTN;
%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*%*
      ARRAY         STDFN[0:15],
                    PKNARRY[0:3],
                    GSA[0:1999],
                    FNAMEPOINTERS[1:14];
      EBCDIC ARRAY  EGSA[0]=GSA[*];
REAL  PROCEDURE DIRREQUEST(FILETITLE,ALLOC,SEGS,LASREC,CDATE
                           ,ADATE,TSTAMP,PARAM);
   ARRAY FILETITLE[0];
   REAL  LASREC,ALLOC,SEGS,CDATE,ADATE,TSTAMP,PARAM;
   BEGIN
      DEFINE
         WAITFORFILEF     =[43:1]#,
         RETAINUSERCODEF  =[42:1]#,
         USERCODEONLYF    =[41:1]#,
         RETURNFULLNAMEF  =[40:1]#,
         DISPLAYFORMNAMEF=[39:1]#,
         ONLYSYSTEMFILESF=[38:1]#,
         RETURNRESIDENTF  =[37:1]#,
         SUBTYPEF         =[15:8]#,
         TYPEF            =[7:8]#,
         MAXCATLEVELF     =[47:8]#,
         ORGLEVELF        =[39:20]#,
         MAXLEVELF        =[19:20]#,
         ERRORF           =[47:1]#,
         ERRORVALUEF      =[46:8]#,
         ADDINFOF         =[46:8]#,
         VALUEF           =[38:6]#,
         SUBVALUE2F       =[38:2]#,
         SUBVALUE3F       =[36:1]#,
         SUBVALUE1F       =[35:3]#,
         SUBVALUE11F      =[35:1]#,
         SUBVALUE12F      =[34:2]#,
         LINKF            =[32:17]#,
         NEXTLEVELLINKF   =[15:11]#,
         RESIDENTSTATEF   =[4:1]#,
         LEVELF           =[3:4]#,
         INFOF            =[15:16]#,
         ONPARTLINKF      =[43:11]#,
         MASK             =1 &
                           1 [7:1] &
                           1 [1:1] &
                           1 [15:1] &
                           1 [11:1] &
                           1 [21:1] &
                           1 [12:1] &
                           1 [2:1] &
                           1 [5:1] &
                           1 [6:1] &
                           1 [14:1] #,
         RCTYPE           =GSA[I+14].[38:4]#,
         RCUNTS           =GSA[I+14].[39:1]#,
         LCMODE           =GSA[I+14].[42:3]#,
         BLKSZ            =GSA[I+2].[47:16]#,
         MAXSZ            =GSA[I+2].[15:16]#,
```

```
        DEFINEEND=#;
        POINTER         S,ST,FTP;
        OWN REAL        I;
        REAL            J,IX,T1,T2,FKIND,NEEDQUOTES;
        BOOLEAN         USERCODE,SYSTEMFILE,B;
        LABEL           EXIT,CONTU;

        FTP:=POINTER(FILETITLE,8);
        IF I = 0 THEN BEGIN
           ST:=POINTER(STDFN,8);
           IF DSPTOSTN(FTP,ST,B) THEN
              IF B:= DISPLAYTOSTANDARD(FTP,ST) THEN BEGIN
                 DIRREQUEST:=REAL(B);
                 GO EXIT;
              END;
           GSA[0]:=200;
           GSA[1]:=0;
           REPLACE POINTER(GSA[200]) BY ST
                    FOR STDFN[0].[47:8];
           B:=GETSTATUS(0 & 1 [42:1] & REAL(B) [38:1] &
                        3 TYPEF & 1 SUBTYPEF,0,MASK,GSA);
CONTU:     IF B THEN BEGIN
              DIRREQUEST:=GSA[1].ADDINFOF;
              GO EXIT;
           END;
        END;
        IF I = 0 THEN I:=2;
        DO BEGIN
           IF I GEQ GSA[0] THEN BEGIN
              I:=0;
              IF GSA[0].ERRORF = 1 THEN BEGIN
                 B:=GETSTATUS(0 & 1 [42:1] & 3 TYPEF &
                              4 SUBTYPEF,0,MASK,GSA);
                 GO CONTU;
              END ELSE
                 DIRREQUEST:=-1;
              GO EXIT;
           END;
           IX:=I;
           IF GSA[I].LEVELF = 0 THEN I:=GSA[I].LINKF+1;
           IF GSA[I].LEVELF = 1 THEN BEGIN
              REPLACE POINTER(PKNARRY) BY " ON ",
                 EGSA[(J:=GSA[GSA[1].ONPARTLINKF].LINKF)+1]
                 FOR REAL(EGSA[J],1),".";
              IF GSA[I].SUBVALUE1F = 3 THEN USERCODE:=TRUE
              ELSE IF GSA[I].SUBVALUE1F = 2
                   THEN SYSTEMFILE:=TRUE;
           END;
           S:=FTP+(FNAMEPOINTERS[GSA[I].LEVELF]);
           SCAN EGSA[(J:=GSA[I].LINKF)+1]
                FOR NEEDQUOTES:REAL(EGSA[J],1) WHILE IN ALPHA;
           IF USERCODE THEN REPLACE S:S BY "("
                       ELSE IF SYSTEMFILE
                            THEN REPLACE S:S BY "*";
           IF FNAMEPOINTERS[GSA[I].LEVELF] GEQ 1 THEN
              IF S-1 NEQ ")" THEN REPLACE S-1 BY "/";
           IF NEEDQUOTES ISNT 0 THEN REPLACE S:S BY """";
```

```
        REPLACE S:S BY EGSA[J+1] FOR J:=REAL(EGSA[J],1);
        IF NEEDQUOTES ISNT 0 THEN REPLACE S:S BY """";
        IF USERCODE THEN REPLACE S:S BY ")"
                    ELSE REPLACE S:S BY "/";
        IF USERCODE OR SYSTEMFILE THEN J:=*+2
                                  ELSE J:=*+1;
        IF NEEDQUOTES ISNT 0 THEN J:=*+2;
        FNAMEPOINTERS[GSA[I].LEVELF+1]:=
           FNAMEPOINTERS[GSA[I].LEVELF]+J;
        USERCODE:=SYSTEMFILE:=FALSE;
        FKIND:=GSA[I].SUBVALUE2F;
        IF IX NEQ I THEN BEGIN
           SEGS:=GSA[I+11]+(IF GSA[I+12]>0 THEN 1 ELSE 0);
           ADATE:=GSA[I+15];
           CDATE:=GSA[I+1];
           TSTAMP:=GSA[I+21];
           PARAM:=GSA[I].ADDINFOF;
           ALLOC:=CASE(RCUNTS*LCMODE) OF (1,1,12,8,6,6)*30;
           ALLOC:=(BLKSZ+ALLOC-1) DIV ALLOC;
           T2:=GSA[I+11] MOD ALLOC*
                 (CASE RCUNTS*LCMODE OF (1,1,12,8,6,6))
               *30+(GSA[I+12]+
                  (CASE RCUNTS*LCMODE OF (47,47,3,5,7,7)))
               DIV (48 DIV
                  (CASE RCUNTS*LCMODE OF (1,1,12,8,6,6)));
           T2:=MIN(T2,BLKSZ)+GSA[I+11] DIV ALLOC*
               (BLKSZ DIV MAXSZ)*MAXSZ;
           LASREC:=(T2+MAXSZ-1) DIV MAXSZ-1;
           IF BOOLEAN(GSA[I+6].[1:1]) THEN
              ALLOC:=(IF SEGS MOD ALLOC = 0 THEN SEGS
                      ELSE (SEGS-(SEGS MOD ALLOC)) + ALLOC)
           ELSE ALLOC:=GSA[I+5]*GSA[I+7];
        END;
        I:=IX+1;
     END UNTIL FKIND NEQ 2;
     REPLACE S-1 BY POINTER(PKNARRY) UNTIL = 4"00";
EXIT:
  END OF DIRREQUEST;
%
  FILE F(KIND=DISK,MAXRECSIZE=14,BLOCKSIZE=420,MYUSE=OUT);
  ARRAY   FILETITLE[0:49];
  REAL    ALLOC,SEGS,LASTR,CDATE,ADATE,TSTAMP,RES,PARAM;
  INTEGER I;
  POINTER P,PP,QQ;
  QQ:=PP:=P:=POINTER(FILETITLE,8);
  REPLACE P BY FILENAME,".";
  REPLACE F.TITLE BY P; OPEN(F);
  REPLACE P BY COMMAND;
     WHILE
        RES:=DIRREQUEST(FILETITLE,ALLOC,SEGS,LASTR,CDATE,
                        ADATE,TSTAMP,PARAM) = 0 DO BEGIN
        SCAN PP:P FOR I:84 WHILE NEQ ".";
        REPLACE PP+1 BY " " FOR 84;
        WRITE(F,<A84>,P FOR 84);
     END;
     PP:=P:=POINTER(FILETITLE,8);
     LOCK(F);
```

```
END SEARCH;
  EXPORT COPY,REMOVE,CHANGE,SEARCH;
  FREEZE(TEMPORARY);
END.
```

APPENDIX F

ÖSYM

OTURUM:1

ÖRNEK SINAV SALON SINAV TUTANAĞI

(14/09/87 14:00)

Salon No.: 10008

Bu Salona gönderilen sınav evrakı:
a) 015 adet Soru Kitapçığı,
b) 015 adet Cevap Kâğıdı,
c) Salon Aday Yoklama Listesi,
d) Sınav Uygulama Yönergesi (yoklama ve denetim işlemlerinden başlayarak sınava ilişkin bütün işlemleri bu Yönergeye uygun olarak yürütünüz.)

İl adı : ADANA Bina No: 01001

Bina adı : ÇUKUROVA ÜNİ.ANFİLER GRUBU BALCALI KAMPÜSÜ

Salon adı : 2.GRUP ANFİLER 4 NOLU DERSLİK

1 Sınav evrakının tümü ya da bir bölümü bu Salona ait değilse, aldığınız önlemleri,

2 Soru Kitapçıkları ve Cevap Kâğıtları Salonda sınava giren adaylar için eksik geldi ise, eksik gelen evrakın türünü ve sayısını,

3 Sayfası eksik, karışık ya da sayfalarındaki Soru Kitapçığı türü harfi farklı olan hatalı Soru Kitapçıkları varsa, aynı türde Soru Kitapçığı ile değiştirdikten sonra, durumu, ilgili adayın numarasını da belirterek aşağıya yazınız.

4 Sınav Tutanağının nasıl doldurulacağı Sınav Uygulama Yönergesinde belirtilmiştir. Raporlandırmaların standart biçimde olması için aşağıda yer alan hususların ilgili olanlarının yanına çarpı işareti konulmalıdır.

☐ Salondaki sınav olaysız ve kurallara uygun olarak geçmiştir.

☐ Salondaki sınava ilişkin raporlandırılması gereken hususlar vardır. Bunlar (aşağıda açıklayınız!)

(Açıklamalarınıza, arka sayfada devam edebilirsiniz.)

Sınava gelmeyen adayların NUMARALARI

Sınava gelmeyen: ___ aday

Sınava gelen: ___ aday

Size bu salonun sınav kapasitesi kaç kişilik tır? ___ kişidir.

( ) Salon sınav için elverişsizdir.

Salondaki sınavın, bu Tutanakta belirtildiği gibi yürütüldüğünü ve Tutanaktaki bilgilerin doğruluğunu beyan ederiz.
(Bu Tutanağı, sınav bitiminde, Cevap Kâğıtlarının en üstüne koyarak mukavva plakalar arasında geri gönderiniz.)

Görevliler	Adı, Soyadı (Ünvanı)	Çalıştığı Kurum	İmza	Saat
Salon Başkanı				
Gözetmen (1)				
Gözetmen (2)				
Gözetmen (3)				
Gözetmen (4)				
Gözetmen (5)				
Gözetmen (6)				
ÖSYM Temsilcisi(*)				

(*) ÖSYM Temsilcisi salona gelmediği takdirde bu kısım imzasız kalacak. Başka bir işlem yapılmayacaktır.

ORNEK SINAV OTURUM:1

YEDEK SINAV EVRAKI KULLANIM ÇİZELGESİ

(14/09/87 14:00)

İLİ.....:ADANA
BİNA NO.:01001

BİNA ADI:ÇUKUROVA ÜNİ.ANFİLER GRUBU
BALCALI KAMPÜSÜ

TANIMI VE KULLANILMA NEDENİ (İlgili göze X koyunuz)									1. Salon Başkanları arasında, eksik gidermek amacı ile sınav evrakı alışverişi yapılmayacaktır. 2. Bozuk ve özürlü sınav evrakı salon sınav evrakından ayrılmaksızın birlikte geri gönderilecektir.				YEDEK SINAV EVRAKI İSTEMİNDE BULUNAN VE EVRAKI TESLİM ALAN SALON BAŞKANININ		TESLİMDE
Cevap Kâğıdı - 1			Cevap Kâğıdı - 2			Soru Kitapçığı			KENDİSİ İÇİN YEDEK SINAV EVRAKI İSTEMİ YAPILAN ÖĞRENCİNİN			Yedek sınav evrakının niçin *istendiğinin* kısa açıklaması	SOYADI, ADI (Varsa, Ünvanı)	İMZASI	SAAT
EKSİK	BOZUK	DİĞER	EKSİK	BOZUK	DİĞER	EKSİK	BOZUK	DİĞER	Salon No.	ÖSYM NUMARASI	SOYADI, ADI				
									← Sınav ertesinde ÖSYM tarafından yapılan sayımlar ve sayımı yapanlar →						Tarih

NOT: Bu Belgeyi Yedek Sınav Evrakı ile birlikte geri gönderiniz.

Bina Sınav Sorumlusu olarak, yukarıdaki bilgilerin doğru olduğunu beyan ederim.

Adı, Soyadı

İmza
(Bina Sınav Sorumlusu)

ÖSYS / SDU - FORM: 533

MİLTEKSAN 209-11-10368

ÖSYM
T.C. YÜKSEKÖĞRETİM KURULU
ÖĞRENCİ SEÇME VE YERLEŞTİRME MERKEZİ

ORNEK SINAV OTURUM:2

SALON TANITIM KARTI

SALON NO.	12341
SALON ADI	ZOOTEKNİ BÖLÜMÜ II ALT KAT DENEME
BİNA NO. VE ADI	06001 A.Ü.ZİRAAT FAK. DIŞKAPI

SALONDAKİ ADAY SAYISI :035
SALONDAKİ GÖREVLİLER : 1 Salon Başkanı ve 2 Gözetmen

ÖSYM - BİB / UYGULAMA - FORM: GENEL - 531

ÖSYM
T.C. YÜKSEKÖĞRETİM KURULU ÖĞRENCİ SEÇME VE YERLEŞTİRME MERKEZİ

ORNEK SINAV
SALON ÖĞRENCİ YOKLAMA LİSTESİ
(17/09/87 10:30)

OTURUM:2
SALON NO: 12341
SAYFA: 1

LİSTE, ÖĞRENCİ NO. SIRASINDADIR. LİSTEDE BULUNMAYAN FAKAT ELİNDEKİ BELGEYE GÖRE BU SALONDA SINAVA GİRMESİ GEREKEN ÖĞRENCİLER SINAVA ALINACAK VE BU ÖĞRENCİLERİN TANITIM BİLGİLERİ LİSTENİN SONUNA EKLENECEKTİR. YOKLAMADA SINAVA GELMEDİĞİ SAPTANAN ÖĞRENCİLER İÇİN BİRİNCİ KOLONA "GİRMEDİ" AÇIKLAMASI YAZILACAKTIR.

SINAV İLİ: ANKARA

BİNA: A.Ü.ZİRAAT FAK. DISKAPI

BİNA NO: 06001

SALON: ZOOTEKNİ BÖLÜMÜ II ALT KAT DENEME

SINAV YOKLAMASINA GELMEYENLER	SIRA NO.	SOYADI	ADI	ÖĞRENCİ NO.
	001	ÖZDEMİR	UFUK	000013
	002	ERTİN	HAKAN	000086
	003	İYİGÜN	MEHMET	000099
	004	ÇAKKALKURT	HÜSEYİN	000117
	005	YELKOVAN	MUSTAFA	000121
	006	ÇAKALLI	H CENGİZ	000272
	007	KÜÇÜKCE	HASAN	000471
	008	ÖZFUTTU	AHMET	000592
	009	CELİK	ENGİN	000636
	010	TELLİOĞLU	CEMAL REŞİT	000653
	011	CATI	AHMET	000701
	012	YAĞLI	DURMUSMURAT	000714
	013	VATANSEVER N	SSEZAİ	000787
	014	ŞEKER	MURAT SEDAT	000791
	015	ERKAN	ERHAN	000805
	016	TALU	HALUK	000882
	017	KURTULUS	İDRİS	000913
	018	TEPE	UĞUR	000926
	019	BAĞISLAR	MESTAN UĞUR	000999
	020	SUR	HAYDAR	001018
	021	ŞEKER	İBRAHİM C	001078
	022	ALTER	ALİ OĞUZ	001082
	023	CANTUTAN	MUSTAFA	001173
	024	MİNRAY	MEMED HAKAN	001217
	025	KARADERELER	SELHAN	001234
	026	CAĞIRAN	ZEKİ	001277
	027	GÜNGÖR	ATAMTÜRK	001281
	028	TAVİLOĞLU	ADNAN KORHAN	001801
	029	KIYAK	MİTHAT	001918
	030	COSKUN	HAMDİ	001922
	031	CİDEMDERE	AHMET HALİT	001965
	032	ŞAHİN	SERDAR	002122
	033	YALÇINER	ALTAN	002269
	034	HÜTEN	MEHMET ALİ	002697
	035	ÖZDEMİR	ŞAHİN	002732

GELMEYENLERİN TOPLAM SAYISI	BU SAYFADAKİ ÖĞRENCİ SAYISI	TOPLAM ÖĞRENCİ SAYISI	SALON BAŞKANININ İMZASI
	035	035	

ÖSYS/SDU-FORM: 537

METEKSAN

ORNEK SINAV OTURUM:2 (17/09/87 10:30)

BİNADAKİ SALONLAR LİSTESİ

BİNA NO :06001

BİNA ADI :A.Ü.ZİRAAT FAK.
DIŞKAPI

SALON NO	ALANI	KAPASİTESİ	ATANAN ADAY SAYISI	SALONUN AÇIK ADI	
12337	01	030	030	DERSANE 5 NOLU	
12340	01	030	007	SU ÜRÜNLERİ BÖLÜMÜ	6 NOLU DERSANE
12341	02	035	035	ZOOTEKNİ BÖLÜMÜ II	ALT KAT DENEME
12342	02	035	003	ZOOTEKNİ BÖLÜMÜ II	ÜST KAT DERSANE
12339	03	050	050	MUHLİS KÜTÜPHANESİ	ÜST SALON
12338	03	075	030	MUHLİS KÜTÜPHANESİ	ALT SALON

ÖSYM

ÖRNEK SINAV

OTURUM: 2

CEVAP KÂĞIDI

① Aşağıdaki yerleri büyük harflerle doldurunuz ve imzalayınız.

Adı / Soyadı : .....

Doğum Tarihi : .....

Adres : .....

.....

.....

İMZA

② Soru kitabı türünü işaretleyiniz.

A ○ B ○

③ ADAYLARIN DİKKATİNE

Cevap kâğıtlarına her adayın adı basılmış durumdadır. Bir hata sonucu başkasına ait cevap kâğıdı size verilmiş olabilir. Bu durumda Salon Başkanına başvurarak adınıza düzenlenmiş cevap kâğıdının bulunmasını sağlayınız. Eğer size ait cevap kâğıdı bulunamazsa ve size boş ya da bir başkasına ait cevap kâğıdı verilirse, aşağıdaki kutuyu işaretleyiniz ve Aday Numarası, Soyadı ve Adı alanlarına kendinize ait bilgileri yazınız.

BU CEVAP KÂĞIDI ADIMA DÜZENLENMİŞ DEĞİLDİR ○

Aday Numarası: 002775

SOYADI: ÖZEN

ADI: SERPİL

④ Aday Numaranızı ayrılan yere önce her kutuya bir rakam gelecek şekilde yazınız. Daha sonra her kutunun altındaki sütunda o rakamın bulunduğu kutucuğu dışarıya taşırmadan karalayınız.

ADAY NUMARASI

0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9

SALON BAŞKANININ DİKKATİNE

Aday sınava girmedi ○

1 Sınava girmeyen adayların cevap kâğıtlarında yukarıdaki kutucuğu kurşunkalemle doldurunuz.

2 Sınav sonunda, sınava giren ve girmeyen tüm adayların cevap kâğıtlarını aşağıdaki Sıra No.suna göre sıraya koyup birlikte paketleyiniz.

Salon No: 10008

Sıra No: 006

01-01021 01-01021

DİKKAT: YUMUŞAK KURŞUNKALEM VE SİLGİ KULLANINIZ. KUTUCUKLARI YANDAKİ ÖRNEK GİBİ KARALAYINIZ.

ÖRNEK: ●

SINAVA GİRECEĞİ ALAN: 1

1	A B C D E	51	A B C D E	101	A B C D E	151	A B C D E
2	A B C D E	52	A B C D E	102	A B C D E	152	A B C D E
3	A B C D E	53	A B C D E	103	A B C D E	153	A B C D E
4	A B C D E	54	A B C D E	104	A B C D E	154	A B C D E
5	A B C D E	55	A B C D E	105	A B C D E	155	A B C D E
6	A B C D E	56	A B C D E	106	A B C D E	156	A B C D E
7	A B C D E	57	A B C D E	107	A B C D E	157	A B C D E
8	A B C D E	58	A B C D E	108	A B C D E	158	A B C D E
9	A B C D E	59	A B C D E	109	A B C D E	159	A B C D E
10	A B C D E	60	A B C D E	110	A B C D E	160	A B C D E
11	A B C D E	61	A B C D E	111	A B C D E	161	A B C D E
12	A B C D E	62	A B C D E	112	A B C D E	162	A B C D E
13	A B C D E	63	A B C D E	113	A B C D E	163	A B C D E
14	A B C D E	64	A B C D E	114	A B C D E	164	A B C D E
15	A B C D E	65	A B C D E	115	A B C D E	165	A B C D E
16	A B C D E	66	A B C D E	116	A B C D E	166	A B C D E
17	A B C D E	67	A B C D E	117	A B C D E	167	A B C D E
18	A B C D E	68	A B C D E	118	A B C D E	168	A B C D E
19	A B C D E	69	A B C D E	119	A B C D E	169	A B C D E
20	A B C D E	70	A B C D E	120	A B C D E	170	A B C D E
21	A B C D E	71	A B C D E	121	A B C D E	171	A B C D E
22	A B C D E	72	A B C D E	122	A B C D E	172	A B C D E
23	A B C D E	73	A B C D E	123	A B C D E	173	A B C D E
24	A B C D E	74	A B C D E	124	A B C D E	174	A B C D E
25	A B C D E	75	A B C D E	125	A B C D E	175	A B C D E
26	A B C D E	76	A B C D E	126	A B C D E	176	A B C D E
27	A B C D E	77	A B C D E	127	A B C D E	177	A B C D E
28	A B C D E	78	A B C D E	128	A B C D E	178	A B C D E
29	A B C D E	79	A B C D E	129	A B C D E	179	A B C D E
30	A B C D E	80	A B C D E	130	A B C D E	180	A B C D E
31	A B C D E	81	A B C D E	131	A B C D E	181	A B C D E
32	A B C D E	82	A B C D E	132	A B C D E	182	A B C D E
33	A B C D E	83	A B C D E	133	A B C D E	183	A B C D E
34	A B C D E	84	A B C D E	134	A B C D E	184	A B C D E
35	A B C D E	85	A B C D E	135	A B C D E	185	A B C D E
36	A B C D E	86	A B C D E	136	A B C D E	186	A B C D E
37	A B C D E	87	A B C D E	137	A B C D E	187	A B C D E
38	A B C D E	88	A B C D E	138	A B C D E	188	A B C D E
39	A B C D E	89	A B C D E	139	A B C D E	189	A B C D E
40	A B C D E	90	A B C D E	140	A B C D E	190	A B C D E
41	A B C D E	91	A B C D E	141	A B C D E	191	A B C D E
42	A B C D E	92	A B C D E	142	A B C D E	192	A B C D E
43	A B C D E	93	A B C D E	143	A B C D E	193	A B C D E
44	A B C D E	94	A B C D E	144	A B C D E	194	A B C D E
45	A B C D E	95	A B C D E	145	A B C D E	195	A B C D E
46	A B C D E	96	A B C D E	146	A B C D E	196	A B C D E
47	A B C D E	97	A B C D E	147	A B C D E	197	A B C D E
48	A B C D E	98	A B C D E	148	A B C D E	198	A B C D E
49	A B C D E	99	A B C D E	149	A B C D E	199	A B C D E
50	A B C D E	100	A B C D E	150	A B C D E	200	A B C D E