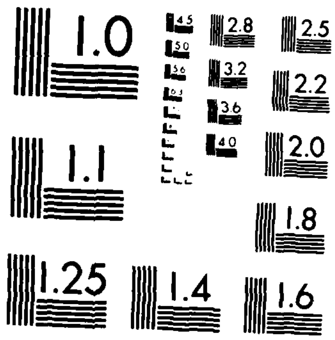
AD-A151 689 COMPUTER ASSISTED FLIGHT SCHEDULE OPTIMIZATION(U) ARMY COMMAND AND GENERAL STAFF COLL FORT LEAVENWORTH KS R D DROWLEY 01 JUN 84 SBI-AD-E750 969 1/1

UNCLASSIFIED F/G 5/9 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

AD-A151 689

COMPUTER ASSISTED FLIGHT SCHEDULE OPTIMIZATION

A thesis presented to the Faculty of the U.S. Army
Command and General Staff College in partial
fulfillment of the requirements for the
degree

MASTER OF MILITARY ART AND SCIENCE

by

ROBERT D. DROWLEY, MAJ, USAF
B.S., United States Air Force Academy, 1971

Fort Leavenworth, Kansas
1984

Distribution unlimited. All material unclassified.

84-3421

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A151689	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER ASSISTED FLIGHT SCHEDULE OPTIMIZATION		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Drowley, Robert D., Major, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Student at the U.S. Army Command and General Staff College, Fort Leavenworth, Kansas, 66027		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS HQ TRADOC, Attn: ATCS-D, Fort Monroe, VA 23651		12. REPORT DATE 1 June 1984
		13. NUMBER OF PAGES 87
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Master of Military Art and Science (MMAS) thesis prepared at CGSC in partial fulfillment of the Masters program requirements, U.S. Army Command and General Staff College, Fort Leavenworth, Kansas 66027.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

COMPUTER ASSISTED SCHEDULING, SCHEDULING, MICROCOMPUTER APPLICATIONS, PASCAL PROGRAMMING

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This project focuses on integrating flying requirements, duties, and training accomplishments on a microcomputer to produce an optimized weekly flight schedule. The thesis includes research for and examination of earlier attempts to solve this problem as well as a detailed explanation of the method used in approaching this work. The main and supporting programs are written in U.C.S.D. Pascal. The programs use a weekly flying schedule, track pilot availability, and integrate training accomplishments to arrive at an optimized flying schedule as prioritized by squadron supervisors.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

MASTER OF MILITARY ART AND SCIENCE

THESIS APPROVAL PAGE

Name of Candidate Major Robert D. Drowley

Title of Thesis Computer Assisted Flight Schedule Optimization

Approved by:

David I. Drummond, Thesis Committee Chairman
Mr. David I. Drummond, MS

Donald F. Hayes, Member, Graduate Faculty
Lieutenant Colonel Donald F. Hayes, MMAS

Clayton W. Freeark, Member, Consulting Faculty
Colonel Clayton W. Freeark, Ph.D.

Accepted this 16th day of May 1984 by Philip J. Brookes, Director, Graduate Degree Program.

The opinions and conclusions expressed herein are those of the student author and do not necessarily represent the views of the U.S. Army Command and General Staff College or any other governmental agency. (Reference to this study should include the foregoing statement.)

COMPUTER ASSISTED FLIGHT SCHEDULE OPTIMIZATION, by Major Robert D. Drowley, USAF, 87 pages.

This project focuses on integrating flying requirements, duties, and training accomplishments on a microcomputer to produce an optimized weekly flight schedule. The thesis includes research for and examination of earlier attempts to solve this problem as well as a detailed explanation of the method used in approaching this work. The main and supporting programs are written in U.C.S.D. Pascal. The programs use a weekly flying schedule, track pilot availability, and integrate training accomplishments to arrive at an optimized flying schedule as prioritized by squadron supervisors.

Table of Contents

CHAPTER ONE

Introduction

Background

Scheduling is one of the most demanding jobs in a tactical fighter squadron. This function is typically manned by three to five officers whose collective responsibilities include:

(1) Producing a weekly by-name schedule while forecasting the next week's requirement.

(2) Forecasting the squadron's mission requirements for subsequent months, quarters, and halves.

(3) Providing the squadron supervisors with a by-day analysis of projected versus actual time flown and events accomplished.

(4) Adjusting the daily and weekly schedules as changes are generated by unforeseen circumstances.

(5) Individually remaining combat ready in the unit aircraft.

Today's aircraft are expensive to buy, maintain, and fly. It is, therefore, incumbent upon the managers of these valuable resources to insure that the training opportunities provided to the aircrews are the best possible. The value of this training is also measured by readiness of our aircrews to perform their primary duty -

that is, go to war should circumstances dictate. Toward that end, it is necessary that all aircrews be as qualified as possible to insure their maximum effectiveness and survivability.

It is the responsibility of the schedulers working with the training personnel to ensure that the best training possible is accomplished. This requires close interface between these functions and the involvement of key supervisors.

Currently there are several styles of scheduling being practiced throughout the Tactical Air Forces. These tend to be very personality dependent with no one style inherently better or worse than another. Some shops are broken into one team which works the current week and another which plans the following. Other squadrons work on the same concept except the team conceiving the plan follows it through to its execution. In another shop, all personnel may work on all aspects of the schedule formulation simultaneously. However, what each style does share with the others is that they are all very time consuming.

The normal scheduling sequence of events is a very time sensitive, labor intensive production to provide the squadron with a quality schedule. If the job simply entailed the matching of aircrews with the flying positions for which they are qualified, there would not be nearly the problems that are encountered with the real world scheduling section. Aircrews have additional duties and other training requirements which compete with their availability to fly. These duties include Supervisor of Flying (SOF), Runway Supervisory Officer (RSO), Squadron Duty Officer (SDO), simulators, alert tours, and required ground training instruction. These events can impact an aircrew's flying availability for periods ranging from hours to days. Add to this the problem of aircrew members who are unexpectedly sick and unable to fly or a higher headquarters mission which must be added onto the schedule, and there is often a dominoing effect as one change cascades to many more throughout the day. It is on these occasions that even the best conceived plan can suddenly be fraught with conflicts.

If these challenges were not enough, it is also the responsibility of the training/scheduling team to keep all of the aircrew members qualified and current in the missions directed by higher headquarters. For them to achieve this monumental task, they must monitor aircrew upgrade training, currencies, crew rest requirements, and numbers of missions accomplished, by type, among other things. Given these requirements, the schedulers must fit their available aircrew members into a schedule which is effected by aircraft maintenance requirements, range space availability, weather, and a myriad of other constraints.

Other major considerations are the type of sorties available and the qualification required to fill them. Aircrews may be qualified to lead or instruct a particular mission or merely fly on the mission as a wingman. Qualification may also be a function of the type mission an aircraft may be capable of flying, e.g. Maverick or Dissimilar Air Combat Training (DACT). Type missions vary from week to week, day to day, or during the day. These may include air-to-ground range missions, air combat maneuvering (ACM), or instruments. Event currencies within these broad categories that must be considered include day and night air refueling, day and night landings, ACM, or air-to-ground weapons events. The consideration and incorporation of these many variables make scheduling a time consuming task.

<u>Problem Statement</u>

The scheduling function of a U. S. Air Force fighter squadron requires the use of many man hours to manage the training resources allocated. This project seeks to develop a computer program to help the scheduler optimize those resources for maximum aircrew training.

<u>Hypothesis Statement</u>

It is hypothesized that an optimized schedule, as prioritized by squadron supervisors, can be produced by integrating the aircrews' training accomplishments, requirements, and qualifications into a schedule-producing computer program.

Purpose

The purpose of this project is to develop a computer program which is compatible with the Cromemco System 2 computer system installed in every U. S. tactical fighter squadron in the world. The program would ease the scheduler's workload while producing an optimized product.

The capability of this program will be demonstrated on an F-15 squadron. This has been because these units currently have a single type mission, air superiority, and deal with one aircrew position, pilot. However, it will be applicable, with minor modification, to crew and multi-role aircraft. Finally, the F-15 was chosen because it is the first unit which has a developed Air Force Operations Resource Management System (AFORMS) which can be integrated into the scheduling program.

Organization

Chapter Two is a review of literature. Previous research and programs dealing with the computer scheduling problem are analyzed. Chapter Three is a description of the computer model. The input parameters and output data are examined. Chapter Four is a detailed explanation of the procedures used in the program and how the program produces its results. Chapter Five contains the conclusion of the study as well as recommendations for further development. Appendix A provides background for the scheduling and training program interface. Appendix B is a depiction of the program's logic flow. Appendix C contains the program listings and a sample of the program output.

real-time capability of knowing the number of missions and events accomplished. AFORMS not only updates the number of events as they are performed, it automatically records the last time the event was accomplished and when it must be done again - if the event is required on a periodic basis.

This capability is an invaluable tool for the squadron supervisors. One of their basic goals is to have all pilots qualified and current in their required missions to the greatest extent possible. This maintains maximum combat readiness and capability.

The AFORMS data base automatically tracks the events accomplished by month during the semi-annual training period. It also computes the number of events remaining by type mission given the level of training being maintained and any proration that was necessary during the half.

From this data, it becomes a matter of finding the pilot who has the greatest requirement to fly a type mission based on the number of missions left as compared to the number of missions required. It is also a relatively simple task to insure that the pilots remain current by monitoring their currency dates. Considering that a squadron flies over 100 sorties a week, this is rarely a problem in the squadron's pool of thirty pilots.

<u>Supervisor Interface</u>

After the primary data has been input, the squadron to insure that an individual is not scheduled for conflicting events simultaneously. Individual deconfliction is a cardinal rule. Should this be violated, either a flight will go unfilled or a critical duty will not be performed.

Another critical area that must be monitored is the length of the duty day. This is a basic requirement of aircrew management as directed by headquarters of the Air Force. Due to the hazardous nature of this profession, regulations state that aircrew members will not perform flight duties over a period of more than twelve hours. All supervisors monitor this requirement closely. A corollary to this restriction is the fact that pilots must have an opportunity for uninterrupted crew rest from the end of one duty day to the beginning of the next. These factors must be carefully adhered to, particularly when a typical squadron starts flying activities around 0400 and finishes after 2100.

It is from these perspectives that the concept of pilot availability must be approached. The program must be able to confirm not only that the pilot is available to fly but that he is not in a crew rest status.

Air Force Operations Resource Management System

AFORMS is a recent innovation to the tracking of flight accomplishments. This system is capable of providing the training and scheduling personnel with a near

IP. This level of qualification will be used to fill those positions as they are required in the scheduling shell. Pilots are also assigned a weather category. This varies from "A", those who may fly to command-directed instrument weather approach minimums, to "E", for those who may not fly in other than clear weather conditions. Weather category can be a selection criteria during periods of marginal weather conditions.

The supervisory status is a function of the type of additional duty that may be performed. For example, the field grade officers who have completed the appropriate instruction are required to supervise the operation of the entire airfield in their capacity as Supervisor of Flying. Lesser qualified officers may be required to fulfill the duty of Squadron Duty Officer which supervises the flying operation at squadron level. Captains and lieutenants are generally qualified as Runway Supervisory Officers. In this capacity, the RSOs are responsible for monitoring the aircraft as they takeoff and land to insure they are properly configured.

Each of these duties are performed for time periods ranging from two to six hours. This severely impacts the pilot's availability to fly when performing these duties.

The final area of discussion is that of the pilot's availability. This function is of critical importance for two reasons. First, a basic function of the scheduler is automation of these unique scheduling requirements is beyond the scope of this program. These types of training missions will be input as "hard lines" at the time that the scheduling shell is input into the computer.

Other missions that will have to be manually inserted into the scheduling shell include VIP flights, checkrides, and rear seat missions in the F-15B model. Missions which are classified as VIP are those which are flown by the senior members of the wing who do not normally fly in the squadron on a continuing basis. Therefore, these individuals would not be included in the pool of available pilots.

Likewise, the pilots who are responsible for administering checkrides are not a squadron resource and would not be in the pool of pilots. Another unique position which can be filled with either pilots or non-rated personnel is the rear seat of the two-seated F-15B. There are only a few specific missions which require that an F-15 qualified pilot fly in that position. These would include instrument checkrides, rear seat qualification for instructor pilots, and when the pilots are receiving their chemical defense orientation flights. This position would be filled as a "hard line" in the weekly schedule.

In the normal scheduling flow, the pilot who is not in an upgrade program has a qualification of MR, EL, FL, or whether the additional duties or flying requirements will be filled first.

Pilot Parameters

The pilot parameters, as indicated in the system description, are made up of several independent entities. Included are the training status, if the pilot is in an upgrade program, flight qualification, weather category, supervisory status, and availability. Each of those categories must be considered during the formulation of a schedule.

The scheduling of the upgrading pilots presents special problems and will be discussed first. In this case, a pilot may be receiving his Initial Qualification Training (IQT) in the aircraft or upgrading from this basic qualification to a combat ready status - Mission Qualification Training (MQT). Other upgrade missions that could be flown by a Mission Ready (MR) pilot could deal with flight qualifications. These are upgrade to element lead (EL) (the ability to lead flights of two aircraft), flight lead (FL) (the ability to lead flights of four aircraft), and the instructor pilot (IP) (the ability to instruct other upgrading pilots). Each of these missions require a specific type of mission (ACM, AAR, etc), must be flown in a specific sequence as demanded by training manuals, and they must be flown with the specific level of supervision (i.e. an IP or squadron supervisor). The

CHAPTER THREE

Model Description

System Description

The goal of this project is to develop a computer program which gives the squadron level schedulers the ability to input timing and duty requirements and to have the computer output an optimized weekly schedule. The foundation of this system is the weekly scheduling shell. This includes the type of flights, with their respective takeoff and landing times, and the other duties (SOF, RSU, SDO, Alert, etc.) which must be filled.

Once the shell has been input, the pilot accomplishments and currencies are read in. These include the number of sorties accomplished/remaining by type mission per individual. This also includes the last time the event was accomplished and the currency requirement that exists, if any.

Next, pilot qualifications are read in. This includes the pilot's training status or level of flight qualification and supervisory status. Another pilot parameter is that of availability. This must be considered to insure a pilot is not scheduled for two events at once.

Finally, the supervisory priorities are input. This enables him to influence the schedule by determining

NOTES

[1]J. P. Arabeyre, J. Fearnley, F. C. Steiger, and W. Teather, "The Airline Crew Scheduling Problem: A Survey," Transportation Science. vol. 3, (1969): 141

[2]Bernardo Nicoletti, "Automatic Crew Rostering," Transportation Science. vol. 9, no. 1, (1975): 34

[3]Carlton L Pannell, Major, USAF. A LINEAR PROGRAMMING APPLICATION TO AIRCREW SCHEDULING. (Ft. Leavenworth, KS: CGSC, 1980).

[4]John P. Wood, Captain, USAF. A MODEL TO EVALUATE F-4E SQUADRON SCHEDULED SORTIE RATES AND PILOT GRADUATED COMBAT CAPABILITY STATUS. (Maxwell AFB, AL: AFIT, 1982).

[5]Brian C. Dugle, Major, USAF. COMPUTER ASSISTED SCHEDULING FOR AIR FORCE TACTICAL FIGHTER SQUADRONS. (Ft. Leavenworth, KS: CGSC, 1983).

[6]A. Geoffry Egge, Major, USAF. COMPUTER SCHEDULING IN A TACTICAL FIGHTER SQUADRON. (Maxwell AFB, AL: ACSC, 1978).

[7]William H. Roege, Captain, USAF. PILOT SCHEDULING IN A FIGHTER SQUADRON. (Massachusetts Institute of Technology: 1983).

[8]Telephone conversation with Major Larry Peters, 120th TFS, Buckley Field, CO, 21 November 1983.

[9]Telephone conversation with Captain Robert Mueller, 479th TFTS, Holloman AFB, NM, 20 November 1983.

[10]Telephone conversation with Captain Glascock, 23d TFW, England AFB, LA, 1 December 1983.

scheduling is done from one day to the next, since one day's schedule is dependent on the number of effective sorties flown on the previous day.

Captain Ray Glascock in the 23d Tactical Fighter Wing has attempted to develop a system which will produce an optimized weekly schedule.[10] At the present, this scheduling program is capable of providing a degree of availability update and deconfliction.

Summary

The requirement for computer assisted scheduling has existed for many years. The development of an algorithm has been attempted at levels ranging from commercial airlines sponsored studies to the efforts of fighter pilots. As the computer systems have evolved, so have the programs to help the scheduling process. As can be seen, much has been done toward developing a deconfliction algorithm. The increasing degrees of success can be attributed to the improvements of the systems and capabilities of the languages used.

pilot accomplishment file, as well as files for availability and requirements. However, this approach was found lacking because it was only developed for flight scheduling. It did not consider ground training events, additional duties, or other constraints to pilot availability.

Operational Developments

Research has uncovered three units which have seriously pursued scheduling since they have had access to computers. One system was run on a main frame computer and two are being developed on the Cromemco microcomputer. One of the two units is pursuing the computer scheduling challenge from an operational point of view, while the other is approaching it from a training unit perspective.

As presented in Maj. Dugal's research, the Air National Guard unit at Buckley Field had developed a workable program. However, after a change of personnel in that unit, the program was judged too inflexible and the squadron has reverted to scheduling by manual methods.[8] In this unit, the schedulers use the computer for tracking accomplishments and currencies.

The 479th Tactical Fighter Training Wing schedulers are developing a program for training unit use.[9] This program will attempt to provide deconfliction and provide continuity between students and instructors as they progress through their training phases. This type of minimum number of sorties, the remaining sorties are distributed to the inexperienced aircrew members.

The aircrew members are prioritized by a pay-off value which was computed for each aircrew member relative to the activity to be performed. The aircrew values for a particular event are compared and the aircrew member with the highest value is selected to perform the event.

The program demonstrated its ability to prioritize a half-day schedule, that is, all aircrew members were considered to fly once. This accomplished the main task of computing and comparing relative values and assigning flight activities. However, the model does not fulfill the needs of a squadron scheduler because deconfliction was not required due to the short period considered. Also, this short time period negated the requirement to consider crew rest. Both requirements are of critical importance and key to successful schedule production.

A second model which has been demonstrated on a microcomputer incorporated many of the important considerations to develop a workable schedule.[7] Included were the development of the training/scheduling interface, aircrew qualifications, crew rest and continuity. The identified objective of this program was to meet the training requirements of the individual aircrew members while minimizing the number of sorties flown.

This program produced a unique pilot data file, requirements. That is, that the proper number of missions, by type, are planned to meet the minimum aircrew requirements for the semi-annual training period.

A third project used the computer to help develop a weekly schedule. This was an interactive approach where the computer listed alternatives for a particular flight activity.[5] The scheduler could then choose an aircrew member based on supervisor guidance. This program took into account pilot qualifications, pilot availability, and event currency. Once a selection was made, the pilot's availability was updated.

This program had developed separate routines for establishing and updating the aircrews' qualifications, event accomplishments, and currencies. It allowed the construction of a weekly schedule which was free of aircrew conflictions. That is, aircrew members were not scheduled to perform two activities at the same time.

Other Approaches

Two other approaches to the fighter squadron scheduling problem addressed the feasibility of implementing an optimized schedule on a micro-computer. The first thesis established an objective of flying aircrew members a minimum number of sorties per week and produced a working program.[6] Additionally, upgrade programs and continuity training were part of the planning factors. Finally, once all aircrew members have flown the affect the fighter squadron scheduler.

One reading addressed the optimization of aircrew scheduling; however, the problem was defined away by stating there was not a "cost" in assigning an airline crew to a rotation. Their only objective was to equalize individual schedules and "employ the personnel in a satisfactory way."[2] In an attempt to equalize the work load distribution, the strategy revolved around a day by day analysis (as opposed to using either aircrews or activities). As in a fighter squadron, however, the model was constrained by the availability of crew members. Their outside activities decreased their availability to fill the flight schedule.

Deconfliction Models

In the past few years, several facets of computer aided scheduling for a fighter squadron have been developed. Two studies have established the desirability and feasibility of using a computer at wing level to develop a six month training sortie allocation.[3] This program would insure that the proper number and distribution of sorties are allocated to satisfy the squadron's training requirements.

Another study developed a program which aids the squadron scheduler in allocating sorties by type.[4] This program takes the semi-annual sortie allotment and insures that the sorties will satisfy the command training analysis.

Airline Crew Scheduling

The problem of producing an optimized schedule has been addressed by the airlines several times. Although somewhat similar, there are many significant differences between the airlines' and fighter squadrons' scheduling requirements.

The similarities include the matching of flight crews with a schedule that must be filled. The scheduler is also faced with constraints of aircrew availability. The airlines must also consider safety regulations, union requirements, and company policies while trying to minimize the cost of operations.

The majority of the papers written acknowledged the significance and difficulty of allocating aircrews to flight rotations, but did not deal with the problem.[1] Rotations are flights that begin at a geographic domicile and the subsequent legs that must be flown to return the aircraft and crew to their original departure point. Therein lies a major difference in attacking the scheduling problem. The airlines' focus was on optimizing the rotations, not the aircrew assignments.

The airlines' foremost problem revolves around returning the aircrews back to their domicile within a pre-defined number of days. Deadheading, a non-productive flight by an aircrew, is a consideration which does not

CHAPTER TWO

Review of Literature

Information Search

The field of computer assisted scheduling has been conceived and developed within the past decade. Most of the previous attempts have been on main frame computers. It has only been within the past five years that specific efforts have been directed to produce a computer aided schedule for a tactical fighter squadron on a microcomputer.

These projects have met with varying degrees of success and will be described as part of this chapter. Industry has not felt compelled to develop this type of program because the nature of an optimized computer aided schedule does not have universal application. There are continuing efforts by military personnel, however, to solve this problem which indicates both its importance and the fact that a suitable solution has not been found.

There are programs currently in the field which solve this problem with varying degrees of success. They range from those whose purpose is to merely print out the schedule to one which accomplishes a degree of deconfliction. Many of these programs have been developed without documentation, however, which hinders a formal leadership provides the final entering argument. This is the relative priority of flying and additional duties. For instance, if several field grade officers are not available for duty, he may wish that that limited resource (i.e. in a senior supervisory capacity) be scheduled first. So the additional duties would be scheduled first to insure that the SOF-qualified officers would be scheduled prior to their availability being impacted by the flying schedule. On the other hand, if a particularly aggressive flying schedule was being contemplated, flights would be run first.

Obviously, if the scheduler discovers that the schedule can not be filled by running one sequence of priorities, he may find that a suitable solution can be found by running a different set of priorities. These type problems could be forecast weeks in advance by trial runs to help the squadron supervisors decide the number and types of personnel that could be released for extended periods of non-availability (e.g. leave or TDY). Predicted shortfalls could also be filled by requesting pilots from outside the squadron's resources. Another option available to the squadron supervisor would be to cut down the length of the duties performed. Though this involves more personnel, the shorter duty periods may fit into the pilot's availability better.

The squadron supervisor can also impact the flow of the squadron duties and flight schedule by limiting the length of the duty day to something less than twelve hours. This is an easy change to implement and it is often used to ease the work load on the squadron personnel. This is significant because, as stated in AFR 60-12, "a secondary goal of operations scheduling is programming aircrew workloads to allow enough time to perform additional duties, take allotted leave, and still have a realistic work schedule."[1]

Program Output

Having completed all of the input actions, the program performs its schedule-filling function. Assuming that the flying schedule is the first priority, the program will search for the first unfilled mission. Once located, the mission type (e.g. DACT, ACM, etc.) and qualification (i.e. FL, IP, etc.) is determined. The AFORMS data is sorted by that type mission to determine who needs to fly that mission the most, based on a requirements ratio. If the pilot is qualified for the position and available, he is assigned to that flying schedule position. His availability and relative need to fly are updated accordingly.

This procedure is repeated until either the entire flying schedule is filled or there are no pilots available to fly in the remaining positions. Once this is accomplished, the additional duty part of the schedule is completed. As described above, the program determines the qualification and time period required, identifies the pilot who has the proper qualification and availability, makes the assignment, and updates his availability. The flying priority to fill these duties could be based on a requirements ratio. This would allow the pilots with a larger requirement to fly to be assigned fewer additional duties and thereby increase their flying availability.

When the program has finished its assignment process, the final result is printed out. As mentioned above, if there are unfilled requirements which must be filled by squadron personnel, the squadron supervisor can exercise his options to either re-prioritize his desires or solicit help from outside of squadron resources.

Changes invariably occur which must also be dealt with. Thus, as pilots become unavailable due to unforeseen circumstances (e.g. Duty Not Including Flying (DNIF) - becomes ill, short-notice TDY, etc.) the schedule must be re-examined or reaccomplished from that time forward to make sure that the changes driven by the loss of one or more pilots do not violate the scheduling constraints.

This may be handled by a manual update for a relatively minor change. This is desirable so the entire week's schedule remains relatively stable. However, if the change requires the movement of one or two key personnel, it may domino throughout the remaining events. In this case, the whole process may be reaccomplished to insure that the integrity of the scheduling guidelines is not violated.

Summary

The schedule process is an involved and highly interdependent process. Given the constraints of the scheduling system, it must integrate inputs from several sources to arrive at a satisfactory solution. As described, this program can solve the scheduling problem on a weekly basis.

NOTES

[1]AFR 60-12, Planning and Scheduling Aircrews and Equipment, para 1 f

CHAPTER FOUR

Program Description

Introduction

This chapter will explain how the program, including its procedures, functions, and supporting programs, was developed. Also, the system, including the hardware and the language used, is described.

The main program integrates the results of several supporting data-generating programs. Since the solution to the scheduling problem revolves around the effective management of the data, the supporting programs will be described first.

Qualification Program

The pilot qualification data generated by this program is currently available in the computer systems at all tactical fighter wings. This program generates the data base which contains the elements described in the paragraph on Pilot Parameters in Chapter Three.

A file of records is produced, with each record containing the pilot's name, flight qualification (MR, EL, FL), weather category (E, D, C, B, A), his flight assignment (A, B, C, D), and a two dimensional array which contains the additional duty type and the number of times that duty was performed. His flight assignment is included for use at a later time if it is desirable to fill missions with pilots from the same flight.

The number of times a person has performed a certain additional duty may be an aid to help equalize the number of additional duties. However, for the purposes of this program, the pilots were selected for the additional duties based on the requirement already described.

AFORMS Program

The AFORMS program is strictly an administrative device to generate the data for the main program. AFORMS is a data base being maintained in the installation level main frame computer. This information reservoir is connected to the squadron by means of the Cromemco computer. The microcomputer system acts as an input device for updating the data base with daily accomplishments, and as an output device to generate a monthly hard copy run for distribution to the pilots.

The Air Force has contracted to have an interface program developed to make this information available to the Cromemco. The expected operational capability date for this is the fall of 1984. When this project is completed, this information will be available to the scheduling program.

The program written to generate the AFORMS-type data does not include the number of events nor the amount of detail that will eventually be available. It does, however, generate the appropriate type of data to run the program and show the capability to incorporate data from this type of file into the scheduling algorithm.

The AFORMS file established by the program consists of records which contain the pilot's name and a two dimensional array which has ten events, listed by name and the number of events required, number accomplished, number remaining, date of last accomplishment, and the date that the event must be accomplished again. If an event does not have a required currency, a date of "999999" is entered so it will not be considered as a constraint.

Schedule Shell Program

The program which establishes the scheduling files produces a separate file for each days' activities. That is, the file contains both the flying and duty activities for each individual day. This is called the schedule shell. The files are set up so that each day may have a varying number of duties or flights. The flight and duty sections of the program should be entered sequentially to facilitate the operations officer's schedule prioritization.

The program prompts the user for entries for each day of the week as well as inputs for the shell. It accepts a flight type (ACBT, DACT, etc) or duty (RSU, SDO, etc), the start/takeoff time of the event, the ending/landing time of the event, the flight position (MR, FL, etc), and the pilots' name if the scheduler has a preference for that particular event. After all flights and duties are entered for one day, the program steps to the next and the process is repeated for each day of the week.

Another routine which prints out the shell (either to the console or printer) works in concert with this program so the inputs can be validated. After the shell is checked for accuracy, the scheduling program can be run to obtain the finished product.

Availability Program

When the availability program is initiated, the entire pilot availability file can be set "true" and then the program accepts manual non-availability inputs. These may be for pilots who are on leave, TDY, extended illness, or who must participate in a specific weekly duty. Examples of this might be the squadron commander's weekly meetings, key staff meetings, or any other commitment which must be filled by a specific individual rather than from a pool of qualified alternatives.

The files can be updated at any time up to and including the time that the main program is run. This allows the flexibility of updating the availability file as requirements become known, on a daily basis, or done totally when the scheduling program is run.

Scheduling Program

The main program consists of several procedures which accomplish key scheduling and administrative functions. The first procedure converts the time of day from the 24 hour military time into a 15 minute time period for use in the availability checking mechanisms. The conversion combines a day of the week computation with the result of a calculation which converts the hours and minutes into 15 minute time periods. Also incorporated is a means of converting the raw start and stop times into times with standard offsets for use in checking and updating pilot availability. These include 2 hour 15 minutes for flight preparation and briefing prior to takeoff and 1 hour 15 minutes after landing for debriefing. Fifteen minutes before and after duties are allowed for travel and duty transition times.

The conversion is necessary because the format of pilot's availability is a time continuum starting at 0001 Sunday and ending 2400 Saturday in 15 minute intervals. These intervals are defined as "true" if the pilot is available or set "false" if he is not. These time intervals are used in conjunction with the procedures which check the pilot's availability, update his availability, check the length of his duty day and set crew rest.

The procedure which checks the pilot's availability simply steps through the time periods encompassed by the events start and stop time, insuring that the pilot is available for each of the time intervals. The update procedure is used after a pilot is selected to perform a duty. This steps through the pilot's availability periods from the start to stop times, turning them all false.

The procedure which checks the length of duty day insures that a pilot is not scheduled for duties for longer than a 12 hour period. It does this by scanning back to the start time of the first duty/flight of the day, adding 12 hours to this time, and then comparing the result to the event stop time. This, of course, assumes that on a day with a later stop time, that the pilot does not report for duty prior to the first duty start time. It also compares the first duty period against his availability for the rest of the duty day to insure he is not conflicting with a previously input "hard line" afternoon duty/flight. This eliminates the possibility of the pilot being put in an out-of-crewrest position later in the day.

To set crew rest for the basis of the following day's activities (that is, no duty may be performed within 12 hours of the previous day's latest stop time), the program scans to the latest stop time or 1600, whichever is later, and sets the pilot's availability "false" for the next twelve hours. This ensures that he will not be considered for a flight or duty within this crew rest period.

Prior to the shell being loaded, the date is entered. For the purpose of this program, the date will be the last day of the week being scheduled. This will facilitate the comparison of currency of the events. The program checks the currencies of all pilots for all events. If a pilot is identified as not current in an event, his name and the associated event will be output for consideration to be put into the weekly schedule as a "hard line" training event. This affords the scheduler the best opportunity to insure the pilot can regain his currency and have the proper level of supervision.

The program searches the shell file for the first unfilled mission. When the type mission is identified, the AFORMS file is sorted to order the pilots by their need to fly that mission type. The program will then compare each pilot in order until it finds the first pilot who has both the proper level of qualification and is available.

The sorting algorithm used in this program was derived from one called Quiksort which was developed by C.A.R. Hoare.[1] The reason that this was chosen over Bubblesort, Shellsort, the insertion variants, and others is its combination of speed and simplicity. The speed of these different algorithms can be compared by examining their relative efficiency when dealing with "N" items and consequently the number of comparisons required to find the solution. The insertion and Bubblesort methods require on the order of N-squared comparisons. Shellsort requires on the order of N to the 1.2 power, while Quiksort requires N log (N) comparisons.[2]

The sorting key by which the algorithm establishes its priorities is based on the pilot's relative need to fly on a by-type mission basis. This need to fly is expressed by the ratio of the number of events accomplished divided by the number of events required. This accounts for the varying requirements of the pilots who are assigned missions according to experienced/inexperienced criteria and Graduated Combat Capability (GCC) level.(See Appendix A for an explanation of GCC). Thus, the people who have relatively fewer sorties accomplished will be considered first for that type mission.

This same type priority assignment can be used for the assignment of additional duties. Considerations must be given to distribute the additional duties equitably. This program selects pilots to perform duties based on their overall relative need to fly, that is, comparing their ratios of total GCC sorties accomplished to their total GCC sorties required.

The squadron supervisor impacts the filling of the schedule by establishing the priority of flying versus the filling of duties. The program can discriminate between the two because the "Flight Position" for additional duties is not applicable and therefore artificially filled with a "0". Using this fact as a discriminator, the file can be broken down into separate flight and duty sections and the appropriate part will be filled first based on the supervisor's priorities.

From this point, the schedule steps through the schedule files, filling the missions/duties in the order specified. The type mission is determined, the sort accomplished, the first qualified and available pilot selected, and his name is placed into that position. Then his availability is updated and his copy of the AFORMS data (not the master AFORMS file) is updated to reflect the accomplishment of that type mission to reduce his relative need to fly. This keeps the same pilot from being in the same list position each time this type flight is sorted.

This process is repeated until all of the schedule files have been addressed. The program will then print out the final result and identify any unfilled position. As mentioned earlier, these may be filled from resources outside the squadron or priorities may need to be realigned to successfully meet all of the tasking.

If, after the schedule is completed, a pilot can not fly his assigned mission, the schedule-producing program can be run again and his position can be annotated as unfilled. From this point, the scheduling program can be run again with the other positions remaining as originally filled, and the computer will attempt to assign a pilot to the position from those who are qualified and required, to accomplish their training events

Also, during this time frame, the wing deconflicts the range requirements with other users of that training space and takeoff times when more than one unit operates from the field. After this has been accomplished, the wing takes the approved squadron inputs and integrates them into a single wing product.

The squadron scheduling section has the responsibilities already delineated as well as the requirement to monitor currencies, crew rest, and the UTE rate for their squadron. The schedulers must be aware of all taskings of the pilots' availability. They must also keep their finger on the day-to-day operation to help resolve unexpected difficulties as they arise during the execution of the schedule.

Additionally, they must deliver the next day's schedule to the squadron operations officer to have it checked and posted in time to have the pilots find their specific flight responsibilities for the following day. All of this must be accomplished while coordinating with Training, Weapons and Tactics, and Standardization / Evaluation. These areas require consultation to insure that their requirements are accounted for in the scheduling process.

Thus, the squadron's scheduling effort is not done in a vacuum. It is inextricably linked to the squadron factors, sympathetic aborts, etc. Once operations and maintenance agree on these factors, the contracted number of sorties can be arrived at.

The culmination of these planning factors is the weekly schedule. This product includes the pilots' names, type missions, takeoff and landing times, and specific aircraft tail numbers. To reach this result, the wing sends the squadrons their requirements for the scheduled week ten to twelve days prior. These requirements include their allocation of training ranges, air refueling block times, static displays, fire department training, etc.

From this point, the operations and maintenance personnel coordinate on specific requirements and capabilities. During this session, such things as takeoff times, aircraft configurations, off-station training, alert commitments, and exercises are discussed. Their cooperation is essential to deal with their common objectives. The number of times aircraft will fly per day which drives the length of maintenances' and operations' duty day results from this meeting.

The product of their combined effort is taken to the next wing scheduling meeting. At this point, the wing commander looks at the successes from the previous week and then the squadrons present their plans for the next. The wing's role during this phase is that of deconfliction of takeoff times and suballocation of ranges, if they are responsible for equitable distribution of the allocated time between the flying squadrons. This will not necessarily be an even distribution since it is based upon the maintenance capabilities and numbers of aircraft assigned per squadron. Similarly, the time may not be equally distributed between the quarters of the year. The wing has the option to adjust the time flow so that it takes into account varying capabilities such as seasonal weather and differences in the number of flying days. The wing must also allow flying time for exercises such as quarterly surges and extended off-base training (e.g., Combined Force Training scenarios and Red Flag operations).

When the wing makes its initial yearly allocations, it publishes its intentions in a yearly plan. This is further refined into more specific quarterly plans. The wing does this with the cooperation of the squadron level schedulers. The squadron schedulers forecast the numbers and types of sorties that will be flown by day.

This plan is refined more by the monthly plan. The monthly plan generates the total number of sorties required to achieve the sorties, hours, and training requirements. The operations schedulers must include attrition factors from maintenance so that the number of effective sorties will match the number required. The attrition factors include losses due to weather, aborts due to maintenance managing operation's taskings (exercises and surges), while insuring that the pilots complete their annual training requirements. All of this must be accomplished with minimum turbulence in terms of human and material resources.

Scheduling System

The wing's scheduling mission is driven by requirements established in AFR 60-12, Planning and Scheduling Aircrews and Equipment, AFR 66-5, Production Oriented Maintenance Organization, and the Tactical Air Command supplements thereto. These regulations dictate the time constraints under which this system must operate. The milestones which will be discussed are based upon the beginning of the week for which the schedule will be implemented.

The wing scheduling division is responsible for the overall wing flying program. It breaks out the yearly hours and sorties requirements from the TAC operations plan (TOP). This document assigns the flying hour and sortie goal for every unit in TAC. The goal is derived by multiplying the number of authorized aircraft per wing by their expected utilization (UTE) rate. The UTE is based upon the amount of monies available for spare parts, number of maintenance personnel, and other factors which determine how often an aircraft can fly.

Once the overall goal is established, the wing is

Appendix A

The Squadron Scheduling Environment

Introduction

The purpose of this appendix is to explain the framework within which the squadron scheduling section must work to effectively perform its mission. It must accomplish its tasks within the constraints of Air Force, Tactical Air Command, and wing regulations while balancing the milestones dictated by operations and maintenance. Included also are the time sensitive demands from pilots involved in training programs and the day to day problems that challenge the schedule at every turn.

What should become evident is that the filling-in of the pilots' names on the empty shell is the last step in a complex continuum which must analyze one week, while forecasting the next, and managing the current. This appendix will highlight the many other duties which must compete for the scheduler's time while they involve themselves in their painstaking schedule-filling duty. If adopted, this schedule-producing program could free the scheduler to allow more time for his other major responsibilities.

The success of a scheduling function is gaged on its ability to meet its primary goals. These include achieving the sortie and hourly goals and successfully

APPENDICES

program more useful and increase the probability that it may be implemented as a helpful tool in the management of fighter squadron resources.

from a hard disk drive rather then the floppy disk drive used.

Since the intent of the program was to demonstrate that an optimized schedule could be produced on a microcomputer, the author attempted to use valid programming methods to achieve reasonable performance. However, this in no way infers that the solution is the only or the most efficient way. In each step of schedule formulation, the author attempted to cut down the selection constraints by the broadest categories first to keep the computer from having to accomplish a detailed analysis on each record.

A variation of the basic program can be produced which alters the standard briefing and debriefing offsets for use when the squadron performs surge operations. In these instances, pilots fly several missions together with minimal time in between flights for briefing and debriefing.

Summary

The basic goal of creating a schedule-producing computer program which is pricritized by the squadron supervisors has been achieved. This has been accomplished by integrating training accomplishments with scheduling requirements.

Having attained this basic goal, the incorporation of recommendations raised in this chapter will make the missions performed. As mentioned before, additional duties accomplished could be brought into the assignment process so that these duties could be equalized by number of duties performed, not just on the number of flights as done in the demonstrated model.

Another area for further consideration includes a capability to check weather minimum qualification. This would be used so that only pilots with a predetermined weather category will be selected to fly when the weather conditions preclude the scheduling of all pilots. Also, for those instances where the sort by accomplishment ratio produces several individuals with an equal need to fly, a further discriminator could be the currency of the pilots in the selected event.

The array of records which tracks the pilot's availability is currently a boolean function, i.e. a person is either available of not. With a minor modification, this could be changed to record why the pilot is not available (e.g. use a character to represent crew rest, flying, duties, alert, etc.). This would facilitate a more accurate representation of pilot availability in the event the schedule had to be selectively re-run.

Observations

The system, as implemented, could accomplish a normal week's scheduling function in under two hours. This would be significantly reduced if the data were accessed attractive.

It must be realized that this program was developed for an operational fighter unit. This fact implies that there are mostly operational continuation training flights being performed on any daily shell with most pilots qualified in most types of missions tasked to the unit. In a training unit, most missions are training with a small percentage (10%-15%) being continuation training. In these units, schedules are accomplished on a daily basis because the production of one day's schedule depends on the successful completion/progression of the missions from the previous day.

Recommendations

As mentioned in the Qualification Program section, there are other ways to influence the selection of the pilots. One may be to fill the schedule flights by members of the same assigned flight (A, B, C, D, etc.). Since this was demonstrated on a single seat aircraft, a modification is required to accommodate multiseat aircraft. The extra crew positions would be easy to add to the shell, but the modification required would be to fly the individuals as formed crews.

Another trivial difference would be the addition of multi-role mission types. Since the mission type requirements are compared to the aircrew qualifications, there are virtually no restrictions to the types of

CHAPTER FIVE

Conclusion and Recommendations

Conclusion

This project was undertaken to demonstrate that an optimized schedule could be produced by using a microcomputer. This involved the integration of training accomplishments with scheduling requirements into the schedule-producing algorithm to produce the optimized product.

The capability was demonstrated using data which were synthetically generated to illustrate the program's ability. Real world data is more restrictive and detailed but the type is the same.

This program will work with the programs developed to support the data types and requirements. However, the overall intent was to use the data available from the installation level data base which resides in AFORMS. When the interface between the Burroughs main frame and Cromemco is accomplished, modifications to records fields will allow the same data to be accessed and the program will produce the same result. If this proves unfeasible, this program will work with its supporting data-producing programs; however, this will entail a dual tracking of training accomplishments and make the implementation less

NOTES

[1]Niklaus Wirth, Algorithms + Data Structures = Programs. (Englewood Cliffs, NJ: Prentice-Hall, Inc., 1976). p. 76

[2]Ibid., p. 85

[3]D.W. Barron, PASCAL-The Language and Its Implementation. (New York, NY: John Wiley & Sons, 1981). p.63

operations are highly interdependent. The program will require a slight modification when run against actual data structures.

available for that time period. If this does not provide a solution, another option is to fill the position from resources outside the squadron, or the entire day may have to be run again.

The System

This program was implemented on an Apple III with 256K memory using U.C.S.D. Pascal. Pascal was chosen for several reasons. It is available on the Cromemco computer and it is an easily portable language. Pascal, which is a derivative of ALGOL-60, is an efficient language with an excellent data structuring capability. Though there is not a standard for speed, programs run in Pascal compare favorably with ALGOL-68 and FORTRAN.[3] Pascal was also chosen because it is easier to program in a higher level language and easier to read for those who would work with the programs in the future.

Summary

The program, as depicted, will produce an optimized schedule. The programs developed can be implemented independently to support the schedule-producing function, however, the intent was to demonstrate the feasibility of using the data generated at base level. The data generated to demonstrate the program's capabilities are more simplistic than those that actually exist, but the general content is the same. This is the crux of the problem because the type of data and the program which performs the training function. It is the responsibility of the training officer to track the training requirements, accomplishments, and insure the smooth progression of the training and upgrade programs.

Whereas the scheduler is interested in ending each quarter "on the time line", the training officer is interested in all of the pilots completing their semi-annual training requirements. These goals must be mutually supportive. The training officer recommends the types of missions to be flown the following weeks based on the types of missions accrued since the start of the training period.

The training officer also recommends the type missions and ranges needed to insure satisfactory continuity for the pilots in the training programs. Additionally, the training officer must help pair the upgrading individuals with their assigned IP.

The basis for his guidance resides in TAC Manual 51-50, Vols. I and VII. TACM 51-50 Vol. I, Flying Training: Tactical Fighter/Reconnaissance Aircrew Training establishes the minimum Air Force standards for training in fighter aircraft. The flying training programs designated in this manual are designed so that the units will achieve highest degree of aircrew combat capability within the available resources. The directive dictates that the training be accomplished in a continuous flow. That is, it is desired that the required events be accomplished throughout the training period rather than in a short period of time.

The guidance for the number of missions required per type mission per pilot is contained in the Graduated Combat Capability (GCC) system. This is an incremental training system which is broken down into three levels. Level A is the basic standard for mission ready pilots. It is the minimum training necessary to remain combat ready in the unit's primary mission. Level B is TAC's recommendation for the distribution on training sorties above those generated for Level A requirements. This level is used for training sorties to increase the proficiency and accomplish specialized tasks not applicable to all pilots. Level C is the sum of all sorties that are tasked against the unit and is the goal for all units (although it is not attainable with current flying time allocations). Included in the GCC accounting system are collateral sorties. These are missions not directly related to combat oriented training. These missions include aircraft proficiency training (instrument sorties and checkrides), deployments, and ineffective GCC sorties. It is expected that slightly over 10% of the GCC level per pilot for each six month period will be collateral.

Continuation Training

Once a pilot has become mission ready he enters not only the GCC sortie system but his flying is also classified as continuation training. This encompasses ground and flying training. Ground training, like flying training, revolves around a six month training cycle. The subjects addressed include Weapons and Tactics Academics, Intelligence/Electronic Warfare, and other flying related areas such as Life Support and physiological training.

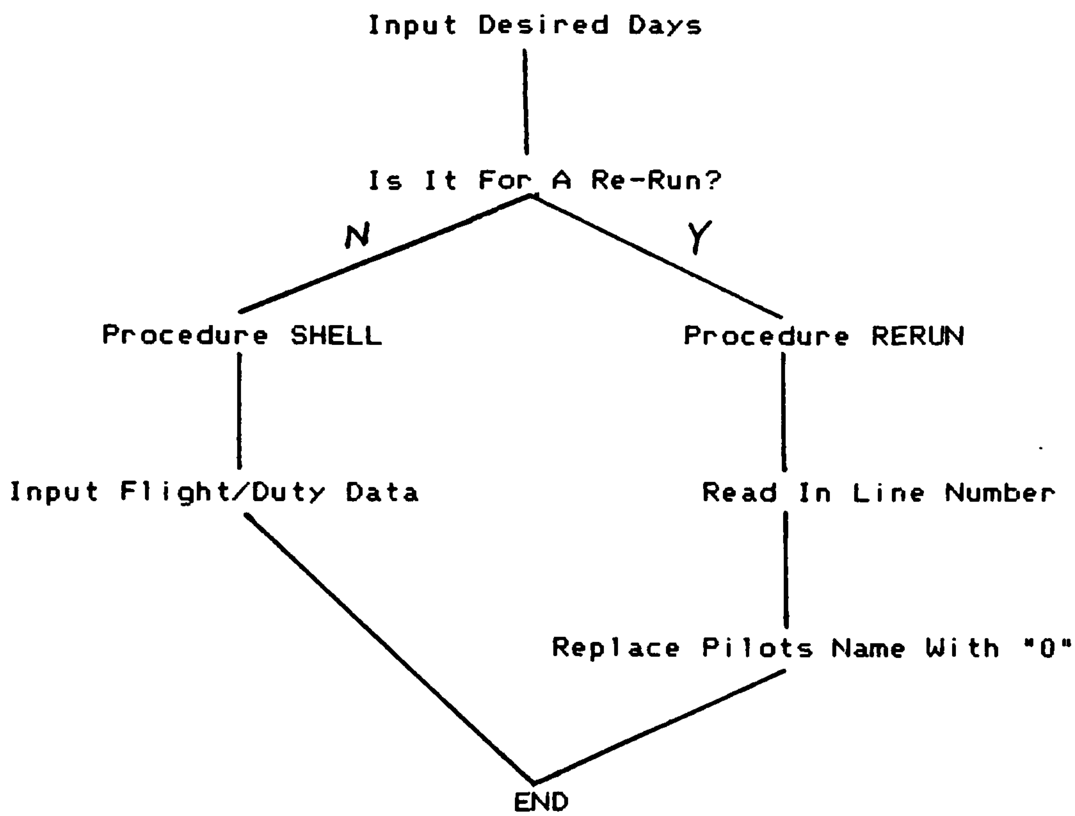
Another area of concentration is that of simulator training. These missions are performed throughout the training period and emphasize such areas as Emergency Procedures, Instrument/Navigation Procedures, and combat profiles which challenge the pilots with wartime rules of engagement, weapons employment procedures, and the tactical use of the Electronic Warfare capabilities.

Finally, the wing is exposed to wartime conditions is quarterly sortie surge exercises. This training exposes both maintenance and pilots to the conditions of rapid turnaround to maximize sortie production in a high pressure environment. The exercise objective is to task the aircrews to participate in tactical scenarios and alert missions at their highest UTE rates. This challenges crews to perform their most realistic missions with limited briefing times and end the mission with a combat turnaround.

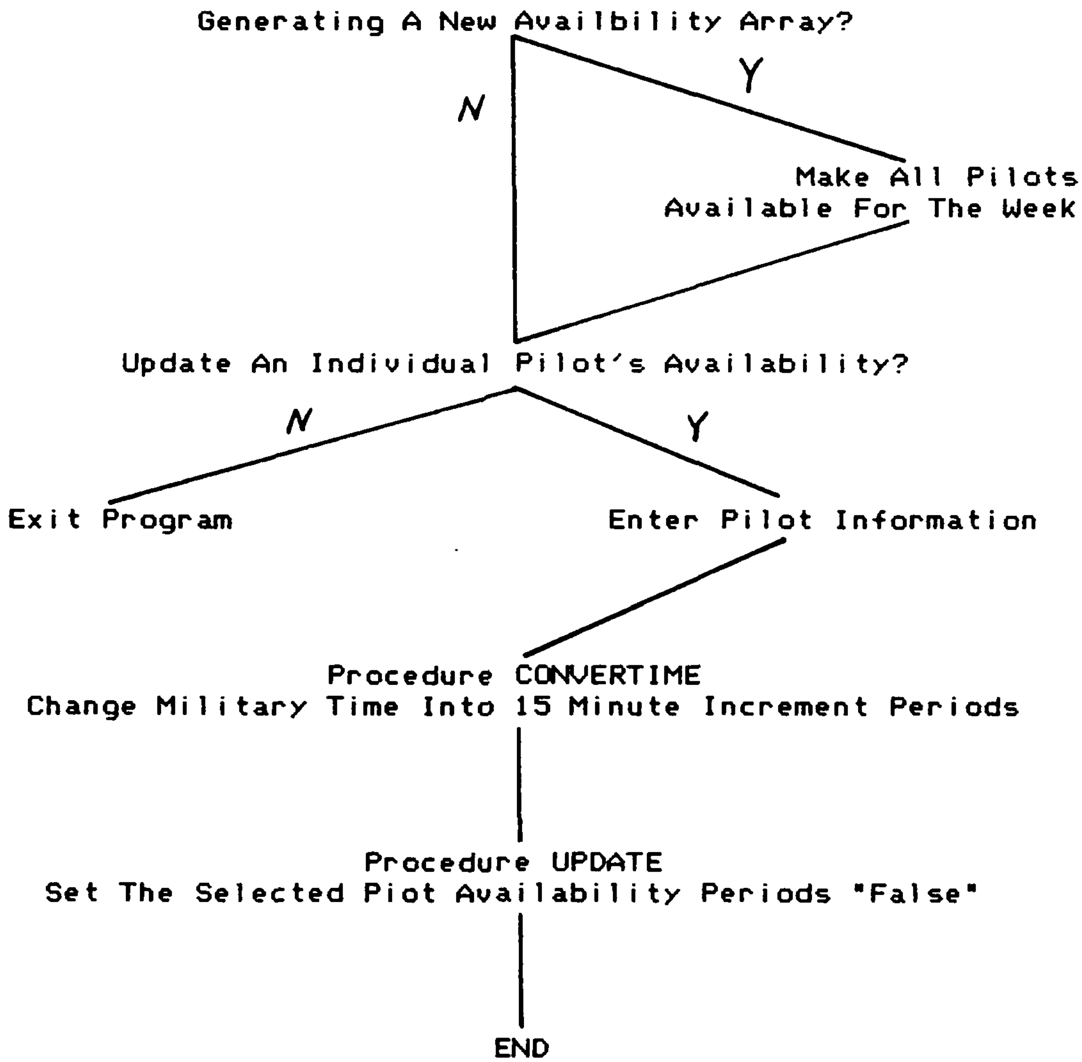
All of these requirements must be tracked by the training personnel. Likewise, these form the basis for the training program which must be scheduled week by week. This is a monumental task which requires daily coordination and adjustment to insure that each pilot has the opportunity to obtain the best training available.

Appendix B

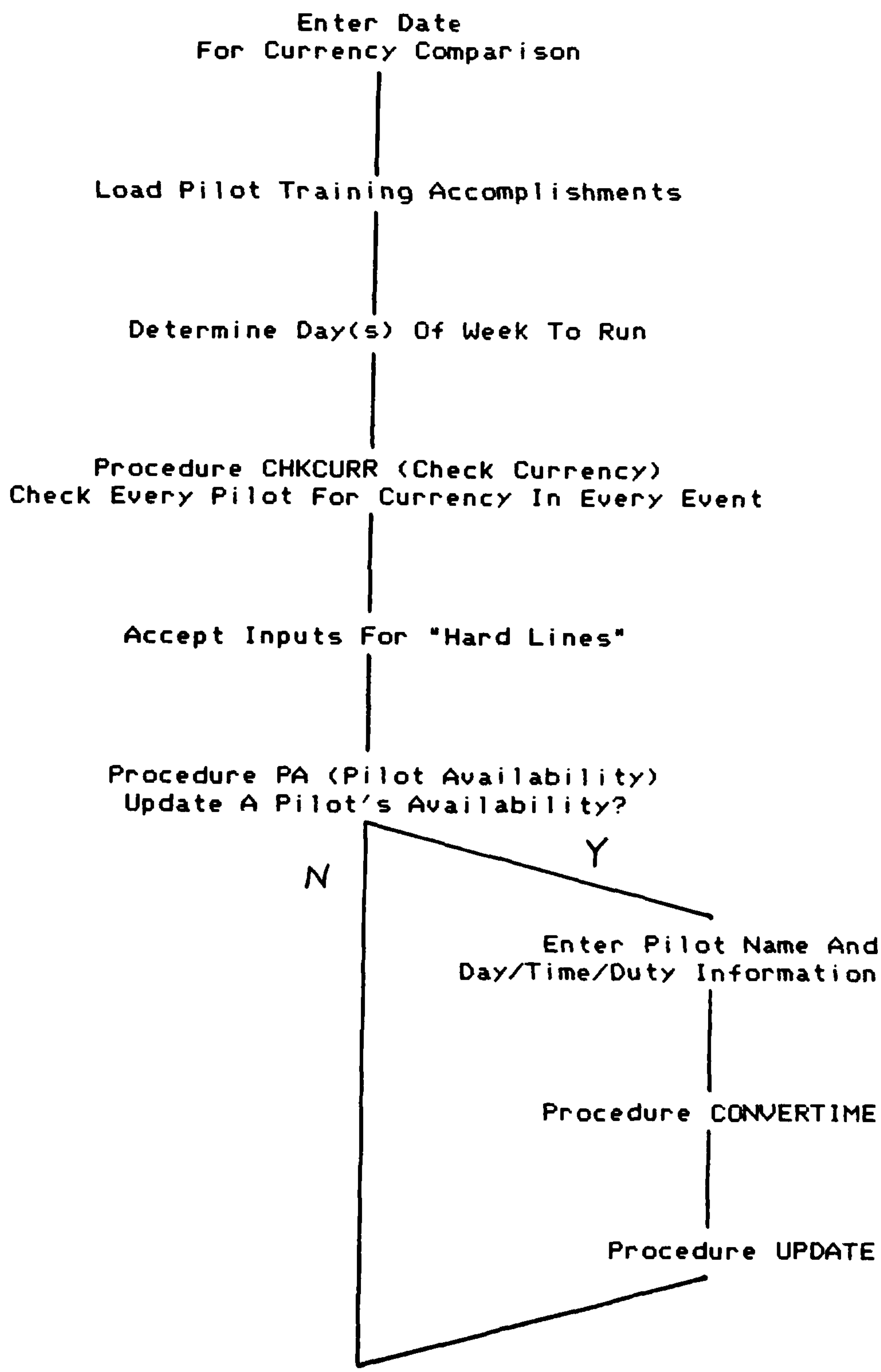
PROGRAM SCHSHEL (SCHEDULING SHELL)



PROGRAM PAUPDATE (PILOT AVAILABILITY UPDATE)



PROGRAM SCHEDULER



Input Supervisor's Fly/Duty Priorities

Step Through Scheduling Shells

Determine Number Of Flights And Duties
For That Schedule Day

Based On Priority Selected
Perform FILLDUTY Or FILLFLY Procedure First

Procedure FILLFLY (Fill Flying Part Of Schedule)

Search Schedule File For Unfilled Flying Lines

Procedure QSORT
Sort Pilot Accomplishment Array Based On
The Type Mission To Be Scheduled

Extract Scheduled Event Start/Stop Times

Procedure CONVERTIME

Search Pilots For Those Qualified To Fill Flight Position

Procedure CHKAVAIL
Insures Pilot Is Available For Duty During The Required Period

Procedure LODD
Insures Pilot Is Not Scheduled Beyond 12 Hour Duty Day

Available and LODD Satisfied?

N

Search For Next Qualified Pilot

Y

Pilot Is Assigned To Shell Flight

Procedure UPDATE

Procedure FILLDUTY (Fill Duty Part Of The Schedule)

Procedure QSORT
Sort On GCC-Accomplishments/GCC-Requirements Ratio

Search For Unfilled Duty

Extract Duty Start/Stop Times

Procedure CONVERTIME

Search For Qualified Pilot

Procedure CHKAVAIL

Procedure LODD

Availability and LODD Satisfied?

N

Search For Next Qualified Pilot

Y

Assign Pilot To Duty

Procedure UPDATE

After Filling Schedule Shell
Search It For Any Unfilled Lines
Print Out Any Unfilled Flights/Duties

Procedure SETCR
Sets Crew Rest From The Latest Duty Time
Or 1600 Whichever Is Later

Next Day's Schedule Is Run

Or

END

APPENDIX C

PROGRAM LISTINGS

The programs listed on the following pages include the main scheduling program and its two supporting programs, Pilot Availability and Scheduling Shell. Any reader who would like to implement these programs is encouraged to contact the author for a copy of the latest version.

```
PROGRAM PAUPDATE;
TYPE
  AVAIL =
    RECORD
      PILOT: STRING[15];
      PER: ARRAY[1..672] OF BOOLEAN END;
VAR
  KEY,ANS : CHAR; I,J,K,L,DAY,START,STOP:INTEGER;
  PAVAIL:FILE OF AVAIL;
  N:STRING[15];
($GOTO+)

PROCEDURE CNVRTIME( VAR DAY,START,STOP: INTEGER);
 LABEL 1;
   VAR  MINT: REAL; TEMP: INTEGER;
 PROCEDURE MIN(VAR TIME:INTEGER);
   BEGIN
     MINT:=(TIME MOD 100)/60;
     IF MINT>0.76 THEN TIME:=4
       ELSE
     IF MINT>0.51 THEN TIME:=3
       ELSE
     IF MINT>0.26 THEN TIME:=2
       ELSE
     IF MINT>0.01 THEN TIME:=1
       ELSE
     TIME:=0
   END;

  BEGIN
    TEMP:=START;  MIN(TEMP);
    START:=(DAY*96)+(START DIV 100)*4+ TEMP;
    TEMP:=STOP;  MIN(TEMP);
    STOP:=(DAY*96)+(STOP DIV 100)*4+ TEMP;
    IF KEY IN['F'] THEN BEGIN START:=START-9;
    STOP:=STOP+5; GOTO 1 END;
    IF KEY IN['D'] THEN BEGIN START:=START-1;
    STOP:=STOP+1; GOTO 1 END;
 1:  KEY:='0'
    END;

 PROCEDURE UPDATE(START, STOP:INTEGER);
 (UPDATES AVAILABILITY OF PILOT FOR SELECTED DUTY TIME)
 VAR J:INTEGER;
   BEGIN
     WRITELN('UPDATE/PAVAIL => ', PAVAIL^.PILOT);
     FOR J:=START TO STOP DO PAVAIL^.PER[J]:=FALSE
   END;
```

```
BEGIN
   RESET(PAVAIL,'/DATABASE/PA.DATA');
   WRITE('ARE YOU GENERATING A NEW WEEKLY PILOT
   AVAILABILITY? ---> ');
   READLN(ANS);
   IF ANS IN['Y'] THEN
   FOR I:=1 TO 30 DO
     BEGIN
(THIS CALLS UP ALL THE PILOT AVAILABILITIES AND CLEARS THEM)
       SEEK(PAVAIL,I-1);
       GET(PAVAIL);
       WITH PAVAIL^ DO FOR J:=1 TO 672 DO PER[J]:=TRUE;
       SEEK(PAVAIL,I-1);
       PUT(PAVAIL)
     END;

 REPEAT
 WRITE('DO YOU WANT TO UPDATE A PILOTS AVAILABILITY? --->  ');
   READLN(ANS);
   IF ANS IN['Y'] THEN
     BEGIN
       RESET(PAVAIL); J:=0;
       WRITE('ENTER THE PILOTS NAME ---> ');
       READLN(N);
       WHILE PAVAIL^.PILOT <> N DO BEGIN J:=J+1;
       GET(PAVAIL) END;
       WRITE('FOR WHICH DAY OF THE WEEK? --->');
       READLN(DAY); DAY:=DAY-1;
       WRITE('ENTER THE EVENT START TIME  --->');
       READLN(START);
       WRITE('ENTER THE EVENT END TIME  ---> ');
       READLN(STOP);
       WRITE('ENTER FLIGHT OR DUTY (F OR D) ---> ');
       READLN(KEY);
       CNVRTIME(DAY,START,STOP);
       UPDATE(START,STOP);
       SEEK(PAVAIL,J); PUT(PAVAIL)
     END
 UNTIL NOT(ANS IN['Y']);
 CLOSE(PAVAIL, LOCK);
 END.
```

DAY ---> 3

MISSION	START	STOP	POSIT	NAME
ACBT	800	920	FL	0
ACBT	800	920	MR	0
ACBT	800	920	EL	0
ACBT	800	920	MR	0
ACBT	820	940	FL	0
ACBT	820	940	MR	0
ACBT	820	940	EL	0
ACBT	820	940	MR	0
DACT	840	950	FL	0
DACT	840	950	MR	0
DACT	840	[illegible]	EL	0
DACT	840	[illegible]	MR	0
BFM	1130	1240	FL	0
BFM	1130	1240	MR	0
DACT	1150	1310	FL	0
DACT	1150	1310	MR	0
DACT	1150	1310	EL	0
DACT	1150	1310	MR	0
ACBT	1210	1320	FL	0
ACBT	1210	1320	MR	0
ACBT	1210	1320	EL	0
ACBT	1210	1320	MR	0
AAR	1500	1715	FL	0
AAR	1500	1715	MR	0
AAR	1500	1715	EL	0
AAR	1500	1715	MR	0
ACBT	1520	1640	FL	0
ACBT	1520	1640	MR	0
ACBT	1520	1640	EL	0
ACBT	1520	1640	MR	0
SOF	1200	1500	0	0
RSU	1100	1400	0	0
SDO	600	900	0	0
DO	900	1200	0	0
DO	1200	1500	0	0
DO	1500	1700	0	0

DAY ---> 1

MISSION	START	STOP	POSIT	NAME
XC	1300	1500	FL	GRAHAM
XC	1300	1500	MR	HARREL
SOF	1300	1500	0	JACKSON

DAY ---> 2

MISSION	START	STOP	POSIT	NAME
ACBT	820	940	FL	0
ACBT	820	940	MR	0
ACBT	820	940	EL	0
ACBT	820	940	MR	0
DACT	850	1000	FL	0
DACT	850	1000	MR	0
DACT	850	1000	EL	0
DACT	850	1000	MR	0
BFM	920	1020	FL	0
BFM	920	1020	MR	0
INST	940	1140	EL	0
INST	940	1140	MR	0
DACT	1200	1310	FL	0
DACT	1200	1310	MR	0
ACBT	1220	1340	FL	0
ACBT	1220	1340	MR	0
ACBT	1220	1340	EL	0
ACBT	1220	1340	MR	0
DACT	1250	1410	FL	0
DACT	1250	1410	MR	0
DACT	1250	1410	EL	0
DACT	1250	1410	MR	0
ACBT	1550	1710	FL	0
ACBT	1550	1710	MR	0
ACBT	1550	1710	EL	0
ACBT	1550	1710	MR	0
ACBT	1620	1740	FL	0
ACBT	1620	1740	MR	0
ACBT	1620	1740	EL	0
ACBT	1620	1740	MR	0
SOF	720	1020	0	0
RSU	800	1100	0	0
RSU	1100	1400	0	0
SDO	1100	1400	0	0
DO	720	1100	0	0
DO	1400	1800	0	0

```
RESET(S,WEEKDAY);

FLY:=0; DTY:=0;
REPEAT
{DETERMINE NUMBER OF FLYING AND DUTY EVENTS EACH DAY}
    IF S^.POSIT <>'0' THEN FLY:=FLY+1 ELSE DTY:= DTY+1;
    GET(S)
  UNTIL EOF(S);

IF PRIORITY=TRUE THEN
  BEGIN
    FILLFLY;
    FILLDUTY
  END
ELSE
  BEGIN
    FILLDUTY;
    FILLFLY
  END;

RESET(S);


 FOR J:=1 TO FLY+DTY DO
   BEGIN
     SEEK(S,J-1);
     GET(S);
     IF S^.PILOT='0' THEN
     WRITELN('DAY ',I+1,' MISSION TYPE: ',S^.MTYPE,
     ' WITH A START TIME:  ',S^.START,' IS UNFILLED');
   END;

 CLOSE(S, LOCK);

 RESET(PAVAIL); (SET CREW REST FOR THE EVENING}
 FOR H:=1 TO 30 DO
   BEGIN
     SEEK(PAVAIL,H-1);
     GET(PAVAIL);
     SETCR(I);
     SEEK(PAVAIL,H-1);
     PUT(PAVAIL)
   END
 END;  (OF STEPPING THROUGH THE WEEK}
CLOSE(PAVAIL,LOCK); CLOSE(PILOTDUTY,LOCK);
END.  {SCHEDULER PROGRAM}
```

```
        CASE I OF
          1: WEEKDAY:='/DATABASE/SUN.DATA';
          2: WEEKDAY:='/DATABASE/MON.DATA';
          3: WEEKDAY:='/DATABASE/TUE.DATA';
          4: WEEKDAY:='/DATABASE/WED.DATA';
          5: WEEKDAY:='/DATABASE/THU.DATA';
          6: WEEKDAY:='/DATABASE/FRI.DATA';
          7: WEEKDAY:='/DATABASE/SAT.DATA'
                         END; { END OF CASE}
        RESET(S,WEEKDAY);
        SEEK(S,J-1); GET(S); S^.PILOT:=N; SEEK(S,J-1); PUT(S);
        FOR X:=1 TO 24 DO WRITELN;
        CLOSE(S, LOCK)
       END
      UNTIL NOT(ANS IN['Y'])
      END; {PROCEDURE CHKCURR}
 BEGIN {SCHEDULER PROGRAM}
 WRITE('ENTER THE END OF THE WEEKS DATE (YYMMDD) --->');
 READLN(DATE);
 {DOWNLOAD AFORMS DATA FROM FILE TO ARRAY FOR SORTING}
 RESET(ADATA, '/DATABASE/AF.DATA');
 FOR I:=1 TO 30 DO
   BEGIN
     SEEK(ADATA,I-1);
     GET(ADATA);
     AD[I]:=ADATA^;
   END;
WRITELN('ENTER DAY OF WEEK TO RUN SCHEDULE:');
WRITE('    SUNDAY --> 1... SATURDAY --> 7         DAY? ---> ');
READLN(D1);
WRITE('FOR HOW MANY DAYS? ---> '); READLN(D2);
   D1:=D1-1;
CHKCURR;
PA;  {MANUAL INPUT OF PILOT NON-AVAILABILITY}
FOR I:=D1 TO (D1+D2-1)  DO   {STEP THROUGH THE DAILY SHELLS}
  BEGIN
    CASE I OF
      0: WEEKDAY:='/DATABASE/SUN.DATA';
      1: WEEKDAY:='/DATABASE/MON.DATA';
      2: WEEKDAY:='/DATABASE/TUE.DATA';
      3: WEEKDAY:='/DATABASE/WED.DATA';
      4: WEEKDAY:='/DATABASE/THU.DATA';
      5: WEEKDAY:='/DATABASE/FRI.DATA';
      6: WEEKDAY:='/DATABASE/SAT.DATA'
                      END;    {END OF CASE}
RESET(S,WEEKDAY);
```

```
   FOR J:=FLY+1 TO DTY+FLY DO
    BEGIN
      SEEK(S,J-1); GET(S);
      IF S^.PILOT='0' THEN
        BEGIN
          START:=S^.START; STOP:=S^.STOP;
          CNVRTIME(I,START,STOP);
          K:=30;
        REPEAT
          IF S^.PILOT='0' THEN
            BEGIN
              RESET(PILOTDUTY);
              RESET(PAVAIL); Z:=0;
   WHILE AD[K].PILOT <> PILOTDUTY^.PILOT DO GET(PILOTDUTY);
   WHILE AD[K].PILOT<>PAVAIL^.PILOT DO BEGIN Z:=Z+1;
   GET(PAVAIL) END;
              FOR H:=1 TO 5 DO
                IF PILOTDUTY^.ADTY[H].ADUTY=S^.MTYPE THEN
                 IF (CHKAVAIL) AND (LODD(I,START,STOP)) THEN
                          BEGIN
                            S^.PILOT:=AD[K].PILOT;
                            SEEK(S,J-1); PUT(S);
                            UPDATE(START,STOP);
                            SEEK(PAVAIL,Z); PUT(PAVAIL);
                            GOTO 2
                          END;
                 K:=K-1
             END UNTIL K=0
        END;  (OF FILLING THE DUTY POSITIONS)
   2:
       END
     END; (FILLDUTY PROCEDURE)
  PROCEDURE CHKCURR;
  BEGIN
   FOR I:=1 TO 30 DO
     FOR J:=1 TO 10 DO
       IF AD[I].EVENT[J].CURR < DATE THEN
         WRITELN(AD[I].PILOT,' IS NON-CURRENT IN ',
         AD[I].EVENT[J].TASK);
  REPEAT
    GOTOXY(10,10);
    WRITE('DO YOU WANT TO INPUT A HARD LINE? ---> ');
    READLN(ANS);
    IF ANS IN['Y'] THEN
      BEGIN
        WRITELN; WRITELN('FOR WHICH DAY OF THE WEEK? ');
WRITE('   SUNDAY --> 1... SATURDAY --> 7      DAY? ---> ');
READLN(I);
        WRITE('FOR WHICH LINE NUMBER? ---> '); READLN(J);
        WRITE('ENTER THE PILOTS NAME ---> '); READLN(N);
```

```
    IF S^.PILOT='0' THEN
      BEGIN
 (DETERMINE WHO NEEDS TO FLY THAT TYPE MISSION THE MOST)
        K:=1;
        REPEAT
          WHILE S^.MTYPE <> AD[1].EVENT[K].TASK DO K:=K+1
        UNTIL S^.MTYPE=AD[1].EVENT[K].TASK;
        QSORT(K);
        START:=S^.START; STOP:=S^.STOP;
        CNVRTIME(I,START,STOP);
          L:=1;
        REPEAT
          IF S^.PILOT='0' THEN
            BEGIN
              RESET(PILOTDUTY);
              RESET(PAVAIL);
              Z:=0;
 (SEARCH PILOT QUAL AND AVAIL ARRAYS FOR SELECTED PILOT)
 WHILE AD[L].PILOT <> PILOTDUTY^.PILOT DO GET(PILOTDUTY);
 (MAKE SURE SELECTED PILOT HAS PROPER QUALIFICATION)
            IF S^.POSIT = PILOTDUTY^.FLTQ THEN
             WHILE AD[L].PILOT <> PAVAIL^.PILOT DO
             BEGIN Z:=Z+1; GET(PAVAIL) END
            ELSE
               BEGIN L:=L+1; GOTO 1 END;
            IF (CHKAVAIL) AND (LODD(I,START,STOP)) THEN
               BEGIN
                 S^.PILOT:=AD[L].PILOT;
 (ASSIGN PILOT TO SHELL)
                 SEEK(S,J-1); PUT(S);
            AD[L].EVENT[K].ACOMP:=AD[L].EVENT[K].ACOMP+1;
                 UPDATE(START,STOP);
                 SEEK(PAVAIL,Z);
                 PUT(PAVAIL);
                 GOTO 2
               END
              ELSE L:=L+1;
              (LOOK AT NEXT PILOT FOR CONSIDERATION)
   1:
          END
        UNTIL L=31
      END; (OF FILLING FLIGHT POSITIONS)
  2:
   END (TO LOOK AT NEXT SHELL FLYING LINE)
   END; (PROCEDURE FILLFLY)

  PROCEDURE FILLDUTY;
  LABEL 2;
  VAR Z:INTEGER;
  BEGIN
    RESET(S);
    QSORT(10);
```

```
 PROCEDURE PA;
   BEGIN
   RESET(PILOTDUTY,'/DATABASE/Q.DATA');
   RESET(PAVAIL,'/DATABASE/PA.DATA');
 (SECTION SOLICITS MANUAL UPDATE OF PILOT AVAILABILITY)
 REPEAT
   WRITELN; WRITELN;
WRITE('DO YOU WANT TO UPDATE A PILOTS AVAILABILITY? --->');
   READLN(ANS); WRITELN;
   IF ANS IN['Y'] THEN
     BEGIN
       RESET(PAVAIL); J:=0;
       WRITE('ENTER THE PILOTS NAME ---> ');
       READLN(N);
       WHILE PAVAIL^.PILOT <> N DO BEGIN J:=J+1;
       GET(PAVAIL) END;
       WRITE('FOR WHICH DAY OF THE WEEK? --->');
       READLN(DAY); DAY:=DAY-1;
       WRITE('ENTER THE EVENT START TIME  --->');
       READLN(START);
       WRITE('ENTER THE EVENT END TIME  ---> ');
       READLN(STOP);
       WRITE('ENTER FLIGHT OR DUTY (F OR D) ---> ');
       READLN(KEY); FOR X:=1 TO 24 DO WRITELN;
       CNVRTIME(DAY,START,STOP);
       UPDATE(START,STOP);
       SEEK(PAVAIL,J); PUT(PAVAIL)
     END
 UNTIL NOT(ANS IN['Y']);
 (GET SQUADRON SUPERVISORS PRIORITY FOR FLYING/DUTIES)
WRITE('ARE FLYING MISSIONS TO BE FILLED PRIOR
                              TO DUTIES? ---> ');
READLN(ANS);
IF ANS IN ['Y'] THEN PRIORITY:= TRUE ELSE PRIORITY := FALSE;
 END; (PROCEDURE PA)
 PROCEDURE FILLFLY;
 LABEL 1,2;
 VAR Z: INTEGER;
 BEGIN
     RESET(PILOTDUTY);
     RESET(PAVAIL);
     RESET(S);
FOR J:=1 TO FLY DO
  BEGIN
    SEEK(S,J-1);
    GET(S);
```

```
 PROCEDURE SETCR(DAY:INTEGER);
 (THIS FINDS THE LAST DUTY PERFORMED ON A DAY AND)
 (SETS NEXT 12 HOURS AS NON-AVAILABLE FOR CREWREST)
 VAR K,J:INTEGER;
   BEGIN
     J:=(DAY+1)*96;
(SETS THE PERIOD COUNTER AT THE END OF THE DUTY DAY)
(NEXT PROCEDURE SEARCHES BACKWARD TO FIND LAST STOP)
(TIME OF THE DUTY DAY OR 1600 WHICHEVER IS LATER)
     WHILE (PAVAIL^.PER[J]=TRUE) AND (J>DAY*96+63) DO J:=J-1;
     FOR K:=J TO J+48 DO PAVAIL^.PER[K]:=FALSE
   END;
 PROCEDURE QSORT(K:INTEGER);
   PROCEDURE SORT (L,R: INTEGER);
     VAR
       A,B: INTEGER;
       X,W: AFORMS;
     BEGIN
       A:=L;
       B:=R;
       X:=AD[(L+R) DIV 2];
       REPEAT

       WHILE AD[A].EVENT[K].ACOMP/AD[A].EVENT[K].RQD <
         X.EVENT[K].ACOMP/X.EVENT[K].RQD DO A:=A+1;
       WHILE X.EVENT[K].ACOMP/X.EVENT[K].RQD <
         AD[B].EVENT[K].ACOMP/AD[B].EVENT[K].RQD DO B:=B-1;
         IF A<=B THEN
           BEGIN
            W:=AD[A];
            AD[A]:=AD[B];
            AD[B]:=W;
            A:=A+1;
            B:=B-1
           END
         UNTIL A>B;
         IF L<B THEN SORT(L,B);
         IF A<R THEN SORT(A,R)
       END; (SORT)
     BEGIN
     SORT(1,30)
     END; (QSORT)
```

```
FUNCTION CHKAVAIL : BOOLEAN;
(THIS FUNCTION INSURES THAT THE PILOT IS AVAILABLE)
(DURING THE ENTIRE DUTY PERIOD)
LABEL 1;
VAR M:INTEGER;
BEGIN
  CHKAVAIL:=TRUE;
  FOR M:=START TO STOP DO
    IF PAVAIL^.PER[M]=TRUE THEN CHKAVAIL:=TRUE
      ELSE
        BEGIN
          CHKAVAIL:=FALSE;
          GOTO 1
        END;
 1:
 END;
 FUNCTION LODD( DAY, START, STOP:INTEGER) : BOOLEAN;
 (FUNCTION INSURES THAT PILOT IS NOT SCHEDULED FOR)
 (A DUTY BEYOND THE END OF HIS 12 HOUR DUTY DAY)
 LABEL 1,2;
 VAR M:INTEGER;
 BEGIN
   LODD:=TRUE;
   M:=DAY*96+1;
   IF STOP>DAY*96+60 THEN
     BEGIN
(CHECKS TIME OF THE END OF PREVIOUS DAYS CREWREST)
       WHILE PAVAIL^.PER[M]=FALSE DO M:=M+1;
       (THIS LOCATES THE BEGINNING OF THE DUTY DAY)
       M:=M+1;
       WHILE M<START DO  IF PAVAIL^.PER[M]=TRUE THEN
       M:=M+1 ELSE GOTO 2;
 2:    IF M+48>STOP THEN LODD:=TRUE ELSE LODD:=FALSE
     END;
   IF START < DAY*96+24 THEN
     BEGIN
(CHECKS LENGTH OF DUTY DAY AGAINST EARLY VS LATE MISSIONS)
       FOR M:=START+48  TO STOP+48 DO
       IF PAVAIL^.PER[M] = FALSE THEN
            BEGIN  LODD:=FALSE; GOTO 1  END
     END;
 1:
 END;
 PROCEDURE UPDATE(START, STOP:INTEGER);
(UPDATES AVAILABILITY OF PILOT FOR SELECTED DUTY TIME)
 VAR J:INTEGER;
   BEGIN
     FOR J:=START TO STOP DO PAVAIL^.PER[J]:=FALSE
   END;
```

```
PROCEDURE CNVRTIME( VAR DAY,START,STOP: INTEGER);
 LABEL 1;
   VAR  MINT: REAL;
   TEMP: INTEGER;
 PROCEDURE MIN(VAR TIME:INTEGER);
   BEGIN

   (THIS CONVERTS THE MINUTES INTO QUARTER-HOUR SEGMENTS)
     MINT:=(TIME MOD 100)/60;
     IF MINT>0.76 THEN TIME:=4
       ELSE
     IF MINT>0.51 THEN TIME:=3
       ELSE
     IF MINT>0.26 THEN TIME:=2
       ELSE
     IF MINT>0.01 THEN TIME:=1
       ELSE
     TIME:=0
   END;


  BEGIN
(START AND STOP EVENT TIMES ARE CONVERTED INTO WEEKLY TIME)
(CONTINUIUM FOR USE IN THE UPDATE OF PILOT AVAILABILITY)
    TEMP:=START;  MIN(TEMP);
    START:=(DAY*96)+(START DIV 100)*4+ TEMP;
    TEMP:=STOP;  MIN(TEMP);
    STOP:=(DAY*96)+(STOP DIV 100)*4+ TEMP;
    IF KEY IN['F'] THEN BEGIN START:=START-9;
    STOP:=STOP+5; GOTO 1 END;
    IF KEY IN['D'] THEN BEGIN START:=START-1;
    STOP:=STOP+1; GOTO 1 END;
    IF S^.POSIT <>'0'  THEN
      BEGIN
(ADJUSTS ACTUAL TAKEOFF AND LANDING TIMES TO ALLOW FOR)
      (BRIEFING AND DEBRIEFING TIME)
        START:=START-9;
        STOP:=STOP+5
      END
    ELSE
      BEGIN
(ADJUSTS START AND END DUTY TIMES TO ALLOW FOR TRAVEL)
      (AND PREPARATION TO PERFORM THE DUTY)
        START:=START-1;
        STOP:=STOP+1
      END;
 1:  KEY:='0'
    END;
```

```
PROGRAM SCHEDULER;

TYPE
  AVAIL =
    RECORD
      PILOT: STRING[15];
      PER: ARRAY[1..672] OF BOOLEAN
    END;
  EVNT = RECORD
      TASK:STRING[6];
      RQD,ACOMP,RMN,LAST,CURR:INTEGER
  END;
  AFORMS = RECORD
      PILOT:STRING[15];
      EVENT:ARRAY[1..10] OF EVNT
  END;
  DUTY = RECORD
      ADUTY:STRING[3];
      PERFORMED:INTEGER
  END;
  QUAL = RECORD
      PILOT:STRING[15];
      FLTQ:STRING[2];
      WX:CHAR;
      FLT:CHAR;
      ADTY:ARRAY[1..5] OF DUTY
  END;

  SCHED = RECORD
      MTYPE:STRING[5];
      START, STOP:INTEGER;
      POSIT:STRING[2];
      PILOT:STRING[15]
  END;
VAR
  KEY,ANS : CHAR;
  DATE,FLY, CNT,DTY,H,I,J,K,L,DAY,START,STOP,
                              D,X,D1,D2:INTEGER;
  PAVAIL:FILE OF AVAIL;
  ADATA: FILE OF AFORMS;
  AD:ARRAY[1..30] OF AFORMS;
  PILOTDUTY:FILE OF QUAL;
  S:FILE OF SCHED;
  PRIORITY : BOOLEAN;
  N:STRING[15];
  WEEKDAY:STRING[18];
  MISSION:STRING[5];
{$GOTO+}
```

```
PROGRAM PRINTSHELL;
TYPE
 SCHED =
   RECORD
    MTYPE : STRING[5];
    START, STOP : INTEGER;
    POSIT : STRING[2];
    PILOT : STRING[15]
 END;
 VAR
   WEEKDAY : STRING[27];
   SCHEDULE : FILE OF SCHED;
   J,K,I: INTEGER;
   P: TEXT;
BEGIN
REWRITE(P,'PRINTER:');
WRITELN('ENTER WEEKDAY FOR HARD COPY OF SCHEDULE: ');
WRITELN('    SUNDAY --> 1');
WRITELN('    MONDAY --> 2');
WRITELN('   TUESDAY --> 3');
WRITELN('WEDNESDAY --> 4');
WRITELN('  THURSDAY --> 5');
WRITELN('    FRIDAY --> 6');
WRITELN('  SATURDAY --> 7');
WRITE('                              DAY? ---> '); READLN(J);
WRITELN; WRITE('FOR HOW MANY DAYS? ---> '); READLN(K);
FOR I:=J-1 TO (I+K-1) DO
  BEGIN
    CASE I OF
      0: WEEKDAY:='/DATABASE/SUN.DATA';
      1: WEEKDAY:='/DATABASE/MON.DATA';
      2: WEEKDAY:='/DATABASE/TUE.DATA';
      3: WEEKDAY:='/DATABASE/WED.DATA';
      4: WEEKDAY:='/DATABASE/THU.DATA';
      5: WEEKDAY:='/DATABASE/FRI.DATA';
      6: WEEKDAY:='/DATABASE/SAT.DATA'
                              END; (END OF CASE)

    RESET(SCHEDULE,WEEKDAY);
    WRITELN; WRITELN; WRITELN;
    WRITELN(P,'                          DAY ---> ',I+1);
    WRITELN(P,' ');
 WRITELN(P,'                           MISSION      START     STOP
    POSIT         NAME');
    WHILE NOT EOF(SCHEDULE) DO
      BEGIN
        WITH SCHEDULE^ DO
   WRITELN(P,MTYPE:26,START:10,STOP:8,POSIT:7,PILOT:14);
        GET(SCHEDULE)
      END;
    CLOSE(SCHEDULE)
  END
END.
```

```
      WITH SCHEDULE^ DO
        BEGIN
    WRITE('ENTER FLIGHT TYPE OR DUTY: --> ');
          READLN(MTYPE);
    WRITE('ENTER TAKEOFF OR START OF DUTY TIME: --> ');
          READLN(START);
    WRITE('ENTER LANDING OR END OF DUTY TIME: --> ');
          READLN(STOP);
    WRITE('ENTER FLIGHT POSITION ("0" IF A DUTY): --> ');
          READLN(POSIT);
          PILOT:='0'
        END;
      PUT(SCHEDULE);
      CLOSE(SCHEDULE, LOCK)
    END;
  UNTIL NOT(ANS IN['Y'])
END; (PROCEDURE RERUN)
BEGIN
  WRITELN('ENTER DAY OF WEEK TO SCHEDULE:');
       WRITELN('   SUNDAY --> 1');
       WRITELN('   MONDAY --> 2');
       WRITELN('  TUESDAY --> 3');
       WRITELN('WEDNESDAY --> 4');
       WRITELN(' THURSDAY --> 5');
       WRITELN('   FRIDAY --> 6');
       WRITELN(' SATURDAY --> 7');
       WRITE('                     DAY? ---> ');
  READLN(J);
WRITE('FOR HOW MANY DAYS?(ENTER "1" FOR A RE-RUN)---> ');
  READLN(K); FOR L:=1 TO 24 DO WRITELN;
FOR I:=J-1  TO (I+ K-1)  DO
  BEGIN
    CASE I OF
      0: WEEKDAY:='/DATABASE/SUN.DATA';
      1: WEEKDAY:='/DATABASE/MON.DATA';
      2: WEEKDAY:='/DATABASE/TUE.DATA';
      3: WEEKDAY:='/DATABASE/WED.DATA';
      4: WEEKDAY:='/DATABASE/THU.DATA';
      5: WEEKDAY:='/DATABASE/FRI.DATA';
      6: WEEKDAY:='/DATABASE/SAT.DATA'
                             END;(OF CASE)
       WRITE('ARE YOU CREATING A NEW SCHEDULE? ---> ');
       READLN(ANS);
   IF ANS IN['Y'] THEN SHELL ELSE RERUN
   END
 END.
```

```
PROGRAM SCHSHEL;
 TYPE
  SCHED=
   RECORD
    MTYPE:STRING[5];  START, STOP:INTEGER;
    POSIT:STRING[2];  PILOT:STRING[15]
   END;
VAR
 WEEKDAY : STRING[27];
 I,J,K,L:INTEGER;
 ANS:CHAR;
 SCHEDULE:FILE OF SCHED;
PROCEDURE SHELL;
BEGIN
WRITELN; WRITELN; WRITELN;WRITELN(WEEKDAY);
 REWRITE(SCHEDULE, WEEKDAY);
WHILE NOT EOF(SCHEDULE) DO GET(SCHEDULE);
REPEAT
 WRITELN; WRITELN;
 WRITE('ENTER A NEW LINE OR DUTY? (TYPE Y OR N): ---> ');
 READLN(ANS);
 IF ANS IN ['Y','y'] THEN
   BEGIN
     WITH SCHEDULE^ DO
       BEGIN
         WRITELN;
       WRITE('ENTER FLIGHT TYPE OR DUTY:'); READLN(MTYPE);
       WRITE('ENTER THE TAKEOFF OR START OF DUTY TIME:');
       READLN(START);
       WRITE('ENTER THE LANDING OR END OF DUTY TIME:');
       READLN(STOP);
       WRITE('ENTER THE FLIGHT POSITION (0 IF A DUTY):');
       READLN(POSIT);
       WRITE('ENTER PILOT NAME, IF KNOWN:'); READLN(PILOT)
       END;
     FOR L:=1 TO 24 DO WRITELN;
     PUT(SCHEDULE)
   END
  UNTIL NOT(ANS IN ['Y','y']);
CLOSE(SCHEDULE,LOCK);
END;
PROCEDURE RERUN;
BEGIN
 REPEAT
    WRITE('ARE YOU UPDATING AN EXISTING DAILY SHELL? ---> ');
    READLN(ANS);
    IF ANS IN['Y'] THEN
      BEGIN
  WRITE('WHICH LINE NUMBER NEEDS TO BE RESCHEDULED? ---> ');
        READLN(L);  RESET(SCHEDULE,WEEKDAY);
        SEEK(SCHEDULE,L-1);
```

DAY ---> 4

MISSION	START	STOP	POSIT	NAME
DACT	800	915	FL	0
DACT	800	915	MR	0
DACT	800	915	EL	BARLOW
DACT	800	915	MR	SEFE
ACBT	820	940	FL	0
ACBT	820	940	MR	0
ACBT	820	940	EL	COL_WINGKING
ACBT	820	940	MR	0
ACBT	840	1000	FL	0
ACBT	840	1000	MR	0
ACBT	840	1000	EL	0
ACBT	840	1000	MR	0
BFM	1130	1230	FL	0
BFM	1130	1230	MR	0
ACBT	1150	1310	FL	0
ACBT	1150	1310	MR	0
ACBT	1150	1310	EL	0
ACBT	1150	1310	MR	0
DACT	1210	1315	FL	0
DACT	1210	1315	MR	0
DACT	1210	1315	EL	0
DACT	1210	1315	MR	0
INST	1500	1700	EL	0
INST	1500	1700	MR	0
DACT	1520	1630	FL	0
DACT	1520	1630	MR	0
BFM	1525	1635	FL	0
BFM	1525	1635	MR	0
DACT	1540	1700	EL	0
DACT	1540	1700	MR	0
SOF	630	930	0	0
RSU	700	1000	0	0
SDO	1300	1700	0	0
DO	700	1000	0	0
DO	1000	1300	0	0

DAY ---> 5

MISSION	START	STOP	POSIT	NAME
ACBT	800	920	FL	0
ACBT	800	920	MR	0
ACBT	800	920	EL	0
ACBT	800	920	MR	0
ACBT	820	940	FL	0
ACBT	820	940	MR	0
ACBT	820	940	EL	0
ACBT	820	940	MR	0
DACT	840	950	FL	0
DACT	840	950	MR	0
DACT	840	950	EL	0
DACT	840	950	MR	0
BFM	1130	1240	FL	0
BFM	1130	1240	MR	0
DACT	1150	1310	FL	0
DACT	1150	1310	MR	0
DACT	1150	1310	EL	0
DACT	1150	1310	MR	0
ACBT	1210	1320	FL	0
ACBT	1210	1320	MR	0
ACBT	1210	1320	EL	0
ACBT	1210	1320	MR	0
AAR	1500	1715	FL	0
AAR	1500	1715	MR	0
AAR	1500	1715	EL	0
AAR	1500	1715	MR	0
ACBT	1520	1640	FL	0
ACBT	1520	1640	MR	0
ACBT	1520	1640	EL	0
ACBT	1520	1640	MR	0
SOF	1200	1500	0	0
RSU	1100	1400	0	0
SDO	600	900	0	0
DO	900	1200	0	0
DO	1200	1500	0	0
DO	1500	1700	0	0

DAY ---> 6

MISSION	START	STOP	POSIT	NAME
ACBT	820	940	FL	0
ACBT	820	940	MR	0
ACBT	820	940	EL	0
ACBT	820	940	MR	0
DACT	850	1000	FL	0
DACT	650	1000	MR	0
DACT	850	1000	EL	0
DACT	850	1000	MR	0
BFM	920	1020	FL	0
BFM	920	1020	MR	0
INST	940	1140	EL	0
INST	940	1140	MR	0
DACT	1200	1310	FL	0
DACT	1200	1310	MR	0
ACBT	1220	1340	FL	0
ACBT	1220	1340	MR	0
ACBT	1220	1340	EL	0
ACBT	1220	1340	MR	0
DACT	1250	1410	FL	0
DACT	1250	1410	MR	0
DACT	1250	1410	EL	0
DACT	1250	1410	MR	0
ACBT	1550	1710	FL	0
ACBT	1550	1710	MR	0
ACBT	1550	1710	EL	0
ACBT	1550	1710	MR	0
ACBT	1620	1740	FL	0
ACBT	1620	1740	MR	0
ACBT	1620	1740	EL	0
ACBT	1620	1740	MR	0
SOF	720	1020	0	0
RSU	800	1100	0	0
RSU	1100	1400	0	0
SDO	1100	1400	0	0
DO	720	1100	0	0
DO	1400	1800	0	0

DAY ---> 1

MISSION	START	STOP	POSIT	NAME
XC	1300	1500	FL	GRAHAM
XC	1300	1500	MR	HARREL
SOF	1300	1500	0	JACKSON

DAY ---> 2

MISSION	START	STOP	POSIT	NAME
ACBT	820	940	FL	GRAHAM
ACBT	820	940	MR	EDIE
ACBT	820	940	EL	DALRYMPLE
ACBT	820	940	MR	BRUENING
DACT	850	1000	FL	HIGHTOWER
DACT	850	1000	MR	CARROLL
DACT	850	1000	EL	BARLOW
DACT	850	1000	MR	WARD
BFM	920	1020	FL	ASHEY
BFM	920	1020	MR	PETERMAN
INST	940	1140	EL	KREMPEL
INST	940	1140	MR	DRAKE
DACT	1200	1310	FL	COLLINS
DACT	1200	1310	MR	DEZONIA
ACBT	1220	1340	FL	JACKSON
ACBT	1220	1340	MR	BEAVER
ACBT	1220	1340	EL	CURRIE
ACBT	1220	1340	MR	HARRELL
DACT	1250	1410	FL	ALLEN
DACT	1250	1410	MR	CARPENTER
DACT	1250	1410	EL	STODDARD
DACT	1250	1410	MR	HALVERSON
ACBT	1550	1710	FL	CHERRY
ACBT	1550	1710	MR	PETERMAN
ACBT	1550	1710	EL	REHM
ACBT	1550	1710	MR	DRAKE
ACBT	1620	1740	FL	ASHEY
ACBT	1620	1740	MR	POOLE
ACBT	1620	1740	EL	KREMPEL
ACBT	1620	1740	MR	MILLS
SOF	720	1020	0	RICE
RSU	800	1100	0	GARBER
RSU	1100	1400	0	LEBRAS
SDO	1100	1400	0	0
DO	720	1100	0	SMITH
DO	1400	1800	0	GARBER

On the previous Day 2, flying missions were filled prior to duties and the 1100 to 1400 SDO did not get filled because all of the SDO-qualified personnel were flying. Day 2 was re-run, filling duties first with the following result. The remaining days were filled by flights first then by duties.

DAY ---> 2

MISSION	START	STOP	POSIT	NAME
ACBT	820	940	FL	GRAHAM
ACBT	820	940	MR	DRAKE
ACBT	820	940	EL	DALRYMPLE
ACBT	820	940	MR	BRUENING
DACT	850	1000	FL	COLLINS
DACT	850	1000	MR	CARROLL
DACT	850	1000	EL	STODDARD
DACT	850	1000	MR	CARPENTER
BFM	920	1020	FL	CHERRY
BFM	920	1020	MR	PETERMAN
INST	940	1140	EL	KREMPEL
INST	940	1140	MR	HARRELL
DACT	1200	1310	FL	ALLEN
DACT	1200	1310	MR	DEZONIA
ACBT	1220	1340	FL	JACKSON
ACBT	1220	1340	MR	EDIE
ACBT	1220	1340	EL	CURRIE
ACBT	1220	1340	MR	BEAVER
DACT	1250	1410	FL	ASHEY
DACT	1250	1410	MR	HALVERSON
DACT	1250	1410	EL	BARLOW
DACT	1250	1410	MR	MILLS
ACBT	1550	1710	FL	CHERRY
ACBT	1550	1710	MR	PETERMAN
ACBT	1550	1710	EL	STODDARD
ACBT	1550	1710	MR	HARRELL
ACBT	1620	1740	FL	RICE
ACBT	1620	1740	MR	POOLE
ACBT	1620	1740	EL	REHM
ACBT	1620	1740	MR	WARD
SOF	720	1020	0	RICE
RSU	800	1100	0	GARBER
RSU	1100	1400	0	LEBRAS
SDO	1100	1400	0	HIGHTOWER
DO	720	1100	0	WARD
DO	1400	1800	0	GARBER

DAY ---> 3

MISSION	START	STOP	POSIT	NAME
ACBT	800	920	FL	ALLEN
ACBT	800	920	MR	BRUENING
ACBT	800	920	EL	DALRYMPLE
ACBT	800	920	MR	BEAVER
ACBT	820	940	FL	JACKSON
ACBT	820	940	MR	EDIE
ACBT	820	940	EL	STODDARD
ACBT	820	940	MR	HARRELL
DACT	840	950	FL	COLLINS
DACT	840	950	MR	CARROLL
DACT	840	950	EL	GARBER
DACT	840	950	MR	WARD
BFM	1130	1240	FL	GRAHAM
BFM	1130	1240	MR	MILLS
DACT	1150	1310	FL	HIGHTOWER
DACT	1150	1310	MR	POOLE
DACT	1150	1310	EL	SMITH
DACT	1150	1310	MR	CARPENTER
ACBT	1210	1320	FL	ASHEY
ACBT	1210	1320	MR	DRAKE
ACBT	1210	1320	EL	CURRIE
ACBT	1210	1320	MR	PETERMAN
AAR	1500	1715	FL	RICE
AAR	1500	1715	MR	HALVERSON
AAR	1500	1715	EL	KREMPEL
AAR	1500	1715	MR	DEZONIA
ACBT	1520	1640	FL	CHERRY
ACBT	1520	1640	MR	EDIE
ACBT	1520	1640	EL	BARLOW
ACBT	1520	1640	MR	WARD
SOF	1200	1500	0	JACKSON
RSU	1100	1400	0	LEBRAS
SDO	600	900	0	HIGHTOWER
DO	900	1200	0	HALVERSON
DO	1200	1500	0	GARBER
DO	1500	1700	0	LEBRAS

DAY ---> 4

MISSION	START	STOP	POSIT	NAME
DACT	800	915	FL	JACKSON
DACT	800	915	MR	MILLS
DACT	800	915	EL	BARLOW
DACT	800	915	MR	SEFE
ACBT	820	940	FL	ALLEN
ACBT	820	940	MR	CARPENTER
ACBT	820	940	EL	COL_WINGKING
ACBT	820	940	MR	DRAKE
ACBT	840	1000	FL	GRAHAM
ACBT	840	1000	MR	BEAVER
ACBT	840	1000	EL	DALRYMPLE
ACBT	840	1000	MR	POOLE
BFM	1130	1230	FL	RICE
BFM	1130	1230	MR	EDIE
ACBT	1150	1310	FL	HIGHTOWER
ACBT	1150	1310	MR	DEZONIA
ACBT	1150	1310	EL	LEBRAS
ACBT	1150	1310	MR	PETERMAN
DACT	1210	1315	FL	ASHEY
DACT	1210	1315	MR	HALVERSON
DACT	1210	1315	EL	KREMPEL
DACT	1210	1315	MR	CARROLL
INST	1500	1700	EL	DALRYMPLE
INST	1500	1700	MR	HARRELL
DACT	1520	1630	FL	COLLINS
DACT	1520	1630	MR	BEAVER
BFM	1525	1635	FL	CHERRY
BFM	1525	1635	MR	CARPENTER
DACT	1540	1700	EL	GARBER
DACT	1540	1700	MR	WARD
SOF	630	930	0	COLLINS
RSU	700	1000	0	GARBER
SDO	1300	1700	0	REHM
DO	700	1000	0	WARD
DO	1000	1300	0	STODDARD

DAY ---> 5

MISSION	START	STOP	POSIT	NAME
ACBT	800	920	FL	HIGHTOWER
ACBT	800	920	MR	BRUENING
ACBT	800	920	EL	REHM
ACBT	800	920	MR	HALVERSON
ACBT	820	940	FL	COLLINS
ACBT	820	940	MR	MILLS
ACBT	820	940	EL	STODDARD
ACBT	820	940	MR	CARROLL
DACT	840	950	FL	RICE
DACT	840	950	MR	EDIE
DACT	840	950	EL	DALRYMPLE
DACT	840	950	MR	DEZONIA
BFM	1130	1240	FL	ALLEN
BFM	1130	1240	MR	BEAVER
DACT	1150	1310	FL	CHERRY
DACT	1150	1310	MR	DRAKE
DACT	1150	1310	EL	KREMPEL
DACT	1150	1310	MR	HARRELL
ACBT	1210	1320	FL	JACKSON
ACBT	1210	1320	MR	PETERMAN
ACBT	1210	1320	EL	SMITH
ACBT	1210	1320	MR	POOLE
AAR	1500	1715	FL	GRAHAM
AAR	1500	1715	MR	WARD
AAR	1500	1715	EL	LEBRAS
AAR	1500	1715	MR	CARPENTER
ACBT	1520	1640	FL	ASHEY
ACBT	1520	1640	MR	EDIE
ACBT	1520	1640	EL	GARBER
ACBT	1520	1640	MR	DEZONIA
SOF	1200	1500	0	RICE
RSU	1100	1400	0	CURRIE
SDO	600	900	0	KREMPEL
DO	900	1200	0	GARBER
DO	1200	1500	0	STODDARD
DO	1500	1700	0	HALVERSON

DAY ---> 6

MISSION	START	STOP	POSIT	NAME
ACBT	820	940	FL	ALLEN
ACBT	820	940	MR	HARRELL
ACBT	820	940	EL	KREMPEL
ACBT	820	940	MR	DRAKE
DACT	850	1000	FL	JACKSON
DACT	850	1000	MR	MILLS
DACT	850	1000	EL	DALRYMPLE
DACT	850	1000	MR	BEAVER
BFM	920	1020	FL	HIGHTOWER
BFM	920	1020	MR	BRUENING
INST	940	1140	EL	GARBER
INST	940	1140	MR	HALVERSON
DACT	1200	1310	FL	ASHEY
DACT	1200	1310	MR	PETERMAN
ACBT	1220	1340	FL	CHERRY
ACBT	1220	1340	MR	CARPENTER
ACBT	1220	1340	EL	CURRIE
ACBT	1220	1340	MR	WARD
DACT	1250	1410	FL	GRAHAM
DACT	1250	1410	MR	POOLE
DACT	1250	1410	EL	LEBRAS
DACT	1250	1410	MR	EDIE
ACBT	1550	1710	FL	JACKSON
ACBT	1550	1710	MR	BEAVER
ACBT	1550	1710	EL	DALRYMPLE
ACBT	1550	1710	MR	BRUENING
ACBT	1620	1740	FL	RICE
ACBT	1620	1740	MR	HALVERSON
ACBT	1620	1740	EL	GARBER
ACBT	1620	1740	MR	CARROLL
SOF	720	1020	0	RICE
RSU	800	1100	0	STODDARD
RSU	1100	1400	0	SMITH
SDO	1100	1400	0	REHM
DO	720	1100	0	CARROLL
DO	1400	1800	0	STODDARD

BIBLIOGRAPHY

Aho, Alfred V., John E. Hopcroft, and Jeffery D. Ullman. The Design and Analysis of Computer Algorithms. Reading, MA: Addison-Wesley Publishing Co., 1974.

Barron, D.W. Pascal - The Language And Its Implementation. New York: John Wiley & Sons, 1981.

Berman, Morton B. Scheduling Aircrews and Aircraft: Problems of Resource Allocation in the Strategic Air Command. #R-1610-PR. Santa Monica, CA: Rand Corporation, 1975.

Chaco, George. Computer-Aided Decision Making. New York: America Elsevier Publishing Co., Inc., 1972.

Date, C.J. An Introduction to Database Systems. Reading, MA: Addison-Wesley Publishing Co., 1981.

DEPARTMENT OF THE AIR FORCE, Headquarters Tactical Air Command. FLYING TRAINING: TACTICAL FIGHTER/RECONNAISSANCE AIRCREW TRAINING. TAC MANUAL 51-50, Volume I, 26 October 1981.

DEPARTMENT OF THE AIR FORCE, Headquarters United States Air Force. Flying: PLANNING AND SCHEDULING AIRCREWS AND EQUIPMENT. AF REGULATION 60-12, 22 March 1979.

Dugle, Brian C. COMPUTER ASSISTED SCHEDULING FOR AIR FORCE TACTICAL FIGHTER SQUADRONS. Ft. Leavenworth, KS: US Army Command and General Staff College, 1983.

Egge, A. Geoffry. COMPUTER SCHEDULING IN A TACTICAL FIGHTER SQUADRON. Maxwell AFB, AL: Air University, 1978.

Gear, C. William. Computer Organization and Programming. New York: McGraw-Hill Book Co., 1969.

Grogono, Peter. Programming in Pascal: Revised Edition. Reading, MA: Addison-Wesley Publishing Company, Inc., 1980, 1978.

Harris, Roy D. Computer Models in Operations Management: A Computer Augmented System. New York: Harper & Row Publishing, 1972.

Knuth, Donald E. The Art of Computer Programming: Vol. 1 Fundamental Algorithms: 2d ed. Reading, MA: Addison-Wesley Publishing Co., 1973, 1968.

______. The Art of Computer Programming: Vol. 2 Seminumerical Algorithms: 2d ed. Reading, MA: Addison-Wesley Publishing Co., 1981, 1969.

______. The Art of Computer Programming: Vol. 3 Sorting and Searching. Reading, MA: Addison-Wesley Publishing Co., 1973.

Lewis, Theodore G. Pascal Programming For The Apple. Reston, VA: Reston Publishing Compnay, Inc., 1981.

Martin, Francis F. Computer Modeling and Simulation. New York: John Wiley & Sons, Inc., 1967.

Martin, James. Computer Data-Base Organization: 2d ed. Englewood Cliffs, NJ: Prentice Hall, Inc., 1975.

Pannell, Carlton L. A LINEAR PROGRAMMING APPLICATION TO AIRCREW SCHEDULING. Ft. Leavenworth, KS: US Army Command and General Staff College, 1980.

Roege, William H. PILOT SCHEDULING IN A FIGHTER SQUADRON. Massachusetts Institute of Technology. 1983.

Stern, Robert A. and Nancy Stern. An Introduction to Computers and Information Processing. New York: John Wiley and Sons, 1982, 1979.

Strunk, Richard R. Can TAC Operations be Computer Scheduled? Maxwell AFB, AL: US Air Force Command and Staff College, 1977.

Wood, John P. A MODEL TO EVALUATE F-4E SQUADRON SCHEDULED SORTIE RATES AND PILOT GRADUATED COMBAT CAPABILITY STATUS. Maxwell AFB, AL: Air University, 1982.

DISTRIBUTION:

Combined Arms Research Library
U.S. Army Command and General Staff College
Fort Leavenworth, KS 66027

Defense Technical Information Center
Cameron Station
Alexandria, VA 22314

Mr. David I. Drummond
U.S. Army Command and General Staff College
Fort Leavenworth, KS 66027

Colonel C. Wayne Freeark
College of Santa Fe
Santa Fe, NM 87501

Lieutenant Colonel Donald F. Hayes
U.S. Army Command and General Staff College
Air Force Section
Fort Leavenworth, KS 66027

Air University Library
Maxwell AFB, AL 36112

HQ TAC/TAFIG
Langley AFB, VA 23665

12th AF/DO
Bergstrom AFB, TX 78743

9th AF/DO
Shaw AFB, SC 29152

1st TFW/DOT
Langley AFB, VA 23665

33rd TFW/DOT
Eglin AFB, FL 32542

49th TFW/DOT
Holloman AFB, NM 88330

474th TFW/DOT
Nellis AFB, NV 89191

23rd TFW/DOT
England AFB, LA 71301

354th TFW/DOT
Myrtle Beach AFB, SC 29577

347th TFW/DOT
Moody AFB, GA 31601

4th TFW/DOT
Seymour Johnson AFB, NC 27531

388th TFW/DOT
Hill AFB, UT 84056

363rd TFW/DOT
Shaw AFB, SC 29152

27th TFW/DOT
Cannon AFB, NM 88101

366th TFW/DOT
Mt. Home AFB, ID 83648

35th TFW/DOT
George AFB, CA 92392

37th TFW/DOT
George AFB, CA 92392

END

FILMED

5-85

DTIC