DIGITAL COMPUTER MULTIPROGRAMMING SYSTEMS

Thesis Submitted by

NAGI MOHAMED ABO EL NAGA

B Sc.

For

M. Sc. Degree

Electrical Department

Faculty of Engineering

Ain Shams University

1973

A C K N O W L E D G E M E N T

I wish to thank Prof. Dr. S.M. Yusuf, Head of the Electrical Engineering Department for his encouragement and support during the execution of this research.

I would like also to express my gratitude and appreciation to Dr. M.A.R. Ghonaimy and Dr. M.R. El Karaksy, for their continuous supervision and helpful guidance through out this work.

Thanks are also due to computer group for much assistance.

Nagi M. Abo El Naga

A B S T R A C T

This thesis deals with multiprogramming (MP) techniques as used with digital computer systems to make efficient utilization of its different resources.

The thesis starts by considering how MP evolved as a cure for situations in which we encounter a mix of highly Input/output - bound and Central Processing Unit-bound programs. Next, MP operating system (OS) architecture is introduced to handle MP environment. Such indispensable topics such as program interrupt system, memory management and scheduling are then introduced.

Storage allocation and the virtual memory concept is thoroughly studied. Paging and segmentation concepts are studied supported by examples of existing real systems; examples included the ATLAS and MULTICS systems.

Scheduling was next augmented as being primarily responsible for the dynamic allocation of resources among a mix of active programs. The different scheduler levels are discussed and different scheduling algorithms are raised. Assignment of priorities and the effect of priorities and the effect of scheduling on file memory operations are also studied.

Finally the architecture of an experimental MPOS is worked out and its constituent parts are thoroughly investigated. Detailed design and flow charts are worked out and successfully implemented on the IBM 1130 computer system. Typical results are shown and the substantial improvement of MP system over a patch processing environment is clearly shown.

C O N T E N T S

Page

Page

I N T R O D U C T I O N

This thesis deals with multiprogramming (MP) techniques as used with digital computer systems make efficent utilization of its different resources.

In chapter (1) evolution of mutiprogramming systems is presented in detail. The situation in which MP systems are effective are those in which I/O - bound and CPU - bound programs are mixed. The structure of operating systems for MP environment are presented and the different modules are described with the job of each module defined. These modules are storage allocator, scheduler, interrupt handler, and job control module.

Chapter (2) deals with storage allocation techniques. The basic problems are first discussed and then evolution of storage allocation is presented starting with the use of base registers and then paging techniques for improving allocation and memory protection. Examples based on existing systems are presented like the "ATLAS" system. Special emphasis was put on the "MULTICS" system for which the segmentation concept was developed in great detail. The hardware support for such system was presented and the advantages of using segmentation was discussed in detail.

Chapter (3) deals with scheduling which is responsible for assigning which programs will take control of the CPU. Different scheduling techniques are presented showing in general the difference between high level and low level scheduling. Time slicing is also presented and dynamic priority assignment techniques are discussed. Details of scheduling for secondary storage devices is presented in details with some specific schemes for drum and disk scheduling. Advantages of scheduling these devices is clearly demonstrated.

Chapter (4) presents an experimental multiprogramming system that was implemented on the IBM 1130 computer of Ain Shams University. The structure of such system took into account the available facilities of this computer which would clearly demonstrate the effectiveness of multiprogramming. Details of the basic moduls of MP operating system were given showing the structure of all required tables, stacks, pointers and indicators. These modules included the interrupt handler, the interrupt identifier, program service subroutines for internal interrupts, I/O service subroutines for external interrupts, and scheduling routines.

Chapter (5) gives the programming details of the above MP operating system together with the necessary flowcharts.

The peripheral equipment used in this system were the card reader and the console printer, and the interrupt key request. Different mixes of programs were tried to illustrate the improvement obtained by using this system over the ordinary batch operating system. Samples of results are included to support the above findings.

CHAPTER 1

Basic Concepts and Multiprogramming Evolution

1.1. Introduction :

To appreciate multiprogramming we have to trace back systems designer's strive for effecient utilization of the different computer components. On the other hand, multiprogramming implementation has brought with it many difficulties which are deeply rooted in operating systems design and memory management philosophy.

In this chapter we first consider Input-Output control to smoothen the effect of slow peripherals as contrasted to the high speed CPU. The fundamental transformation towards efficient utilization of CPU time is the overlapping of CPU and I/O operations. There are various techniques for achieving overlap, and various tradeoffs have been made in order to achieve it.

Buffering techniques, are considered first; then we come to the concept of channels.

Basically, a channel provides a path for data and control signals to be passed. At the CPU interface one finds a variation in the amount of support that the CPV provides for I/O operation, the extent of true independence of channels from the CPU and from each other and the degree of the hardware supporting channel operations. In the presence of an I/O controller, the necessary

information is sent to the channel thus making it to go ahead under its own steam. A channel program is then executed by the channel, independent from the CPU.

For highly I/O bound programs, we notice that even when the program runs in its hypothetical minimum time, the CPU is still very lightly utilized. This is where multiprogramming is very tempting. Multiprogramming is the phenomenon of having more than one program "active" at a point in time, any of which if given the control of CPU can make use of it.

Next in this chapter, multiprogramming operating systems are considered. The reason for considering operating systems is that in view of multiprogramming systems an appreciation of the cost in system features and services to support the multiprogramming capability is in order .

Of particular importance to a multiprogramming machine is the interrupt. A full description of the multichannel, multiprogramming asynchronous computer system is impossible without the support of an interrupt system. This is also explained in this chapter.

Dynamic storage management and the impact of multiprogramming is also considered.

A fundamental goal of store management is to achieve a balance between supply and demand for core and to provide mechanisms to place things in core when they are needed there. The problem is complicated more and more as deeper hierarchies of storage are encountered, as it is the case for real systems. The multiprogramming environment has, by its very nature, contributed to the interest in dynamic storage allocation. The effectiveness of multiprogramming as a technique for increasing the effective utilization of components of a computer system in part depends upon the range of choice of programs that may be run together and upon the overhead the running together of programs implies.

Finally, in multiprogramming environment, effective scheduling is vital. In essence, multiprogramming involves the simultation of a number of processors from one processor. Questions of priority and precedence are there, but in contrast to scheduling of a sequential system, questions of effective usage and utilization of individual components are strongly present.

1.2. Input-Output Control :

In early computer systems, programmers were allowed to operate the computer by themselves or an operator manually sequenced the jobs for the computer.