UNIT 2: PROCESS SCHEDULING

Unit Structure

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- Concept of process scheduling
- Idea about primitive operating systems
- Basic of process states
- Concept of parallel processing
- Brief on scheduling algorithms

2.1 Introduction

Maximum systems have a great figure of processes with abrupt CPU bursts bracketed between I/O requests as well as a little figure of processes with elongated CPU bursts. To allow good time-sharing behaviour, we may pre-empt a moving process to allow another one flow. The arrange list, additionally comprehended as a run chain, in the operating system preserves a history of complete processes that are eager to run moreover not blocked on input/output or another blocking system demand, alike as a semaphore. The entries in this document are pointers to a procedure control block, which accumulates all information besides state about a process.

When an I/O approach for a process is accomplish, the process behaves from the waiting state to the ready state further acquires placed on the run chain.

The process scheduler is the constituent of the operating system that is concerned for adopting whether the recently running process should extend running moreover, if not, which process should flow next. There are four conditions that may happen where the scheduler needs to place in too make this decision:

The fresh process flows from the running to the waiting condition due to it issues an I/O request or numerous operating system demand that cannot be satisfied currently. The recent process halts.

A timer interrupt drives the scheduler to run as well as decide that a process acquires run for its allocated duration of time as well as it is time to proceed it from the active to the develop state.

An I/O operation is accomplished for an approach that demanded it besides the process here and now moves from the halting to the warm up state. The scheduler may that time choose to pre-empt the currently-running process as well as move this newly-ready process into the running state.

A scheduler is a pre-emptive scheduler if it acquires the aptitude to acquire invoked by an interrupt as well as delivers a process out of a moving state to allow another process flow. The last two events in the furthermost list may drive this to occur. If a scheduler cannot abduct the CPU elsewhere from a process that time it is an adaptable or non-pre-emptive scheduler. Primitive operating systems like as Microsoft Windows 3.1 or Apple Mac OS following to OS X are examples of cooperative schedulers. Older batch processing systems adhered run-to-completion schedulers where a mechanism raced to abandonment before numerous foreign processes would be assigned to run.

The judgments that the scheduler brings about concerning the sequence as well as length of time that mechanisms may run is designated the scheduling algorithm (or scheduling policy). These judgments are not contend ones, as the scheduler acquires only a restricted number of information about the processes that are develop to run. An excellent scheduling algorithm should:

Be attractive – allocate each process a pretty share of the CPU, permit each process to proceed in a feasible measure of time.

Be accommodating – preserve the CPU busy whole the time.

Enlarge throughput – service the largest feasible notation of jobs in an allotted measure of time; decrease the measure of time user's essential wait for their outcomes.

Abbreviate response time – collaborative users should inspect good performance.

Be predictable – an allotted job should appropriate about the equal number of time to run when run multiple times. This preserves users realistic.

Minimize overhead – don't excrete too many means. Keep approximating time as well as context switch time at a minimal.

Maximize resource utilize – contribute processes that will utilize underutilized means. There are two causes for this. Maximum devices are sluggish matched to CPU actions.

2.2 Process States

The process state complying of all imperative to begin again the process accomplishment if it is somehow laid monologue impermanent. The process state consists of at least resulting:

- Code for the program.
- Programs fixed data.
- Program's active data.
- Program's approach call stack.
- Contents of general purpose register.
- Contents of program counter (PC)
- Contents of program status word (PSW).

- Operating Systems resource in application.

A process flows through an arrangement of different process states.

- New State: The process being created.
- Running State: A process is discussed to be running if it holds the CPU, that is, process accurately employing the CPU at that definite condition.
- Blocked (or halting) State: A process is lectured to be blocked if it is halting for several conditions to occur like that as an I/O achievement preceding it can precede. Record that a process is unable to flow until several external condition occurs.
- Ready State: A process is discussed to be ready if it utilizes a CPU if one were suitable. A ready state process is run able however transiently stopped flowing to allow another approach run.
- Terminated state: The process seizes finished execution.

Check your progress 1

1. In a New State, the starts _______.

   a. developing c. writing

   b. reading d. all

2. The CPU is used in ________state.

   a. new c. halting

   b. running d. none of these

2.3 Virtual Processor

A virtual CPU (vCPU) additionally recognized as a virtual processor, is an actual central processing unit (CPU) that is allocated to a virtual machine (VM).By shortfall, virtual machines are assigned one vCPU each. If the actual host acquires multiple CPU cores at its desertion, nevertheless, that time a CPU scheduler allocates completion contexts as well as the vCPU centrally serves a series of duration slots on logical processors.

Since processing time is billable, it is notable for an administrator to comprehend how his cloud donator documents vCPU application in an invoice. It is additionally important for the administrator to determine that accumulating more vCPUs will not automatically advance action. This is due to as the notation of vCPUs flows up, it serves increased complicated for the scheduler to arrange time slots on the real CPUs, along with the wait time can disgrace performance.

In VMware, vCPUs are component of the symmetric multi-processing (SMP) multi-threaded approximate model. SMP additionally allocates threads to be break across multiple actual or feasible cores to alter performance of additional parallel virtualized works. vCPUs allow multitasking to be acted consecutively in a multi-core ambience.

(1) In virtualized server surroundings, a virtual processor is a CPU core that is apportioned to a virtual machine. There can be additional virtual processors allocated than real cores feasible, which allocates virtual machines to participate the equivalent core.

(2) In parallel processing surroundings that adheres more data components than processors, a virtual processor is a duplicated processor. Virtual processors conduct in series, not in parallel, although authenticate applications that need a processor for each data component to flow in a computer with fewer processors.

Check your progress 2

1. A virtual processor is a ________core.

a. hard disk	c. memory
b. CPU	d. none

2.4 Interrupt Mechanism

An interrupt is a signal from equipment affixed to a computer or from an approach within the computer that brings about the core program that conducts the computer (the operating system) to stop as well as figure out what to conduct next. Almost entire personal (or larger) computers here and now are interrupt-driven - that is, they begin down the index of computer instruction s in one program (maybe an application like as a word processor) further preserve running the instructions until both

1. Actuate any further
2. Interrupt signal is detected

Following the interrupt signal is perceived, the computer either begins again running the program it endured running or commences running another program.

Acutely, an individual computer can function only one computer instruction at an interval. Although, since it can be delayed, it can acquire turns in which programs or sets of instructions that it functions. This is comprehended as multitasking. It assigns the user to execute a enumerate of contradictory things at the same time. The computer clearly acquires turns managing the programs that the user consequentially starts. Of course, the computer conducts at speeds that generate it seem as although all of the user's works are being acted at the same time. (The computer's operating system is beneficial at using compact pauses in operations besides user think time to work on external programs.)

An operating system usually acquires several code that is identified an interrupt handler. The interrupt handler prioritizes the interrupts as well as preserves them in a chain if more than one is halting to be handled. The operating system acquires another brief program, sometimes termed a scheduler, which circumscribes away which program to assign control to next.

In common, there are hardware interrupts as well as software interrupts. A hardware interrupt arises, for exemplary, when an I/O operation is accomplished like as reading some data into the computer from a tape drive. A software interrupt arises when an application program ceases or appeals assured services from the operating system. In a personal computer, a hardware interrupt request (IRQ) acquires a value affiliated with it that associates it with a definite device.

Five conditions must be true for an interrupt to be generated:

1) Device arm,
2) NVIC enables,
3) Global enable,
4) Interrupt priority level must be higher than current level executing, and
5) Hardware event trigger.

Check your progress 3

1. Interrupt mechanism uses__________.

   a. one programs   c. many programs

   b. two programs   d. all

2. Which is not a valid condition for an interrupt to be generated?

   a. device arm   c. global enable

   b. NVIC enable   d. interrupt priority should be low

2.5 Scheduling Algorithms and Its Performance

Previously when a set of preference relations for a project is known, then the necessary scheduling trouble turn out to be formation of a Priority List. There are a group of potential strategies to facilitate which lead to formation of a Priority List. At this time, we will think only two of these strategies:

- Decreasing-Time Algorithm
- Critical-Path Algorithm

Decreasing-Time Algorithm

Decreasing-Time Algorithm (DTA) is based on simple strategy:

Perform the longer jobs initially as well as save the shorter jobs for final. Basically it places the DTA to make a Priority List by listing the everyday jobs in declining order of dispensation times. Tasks through equal processing times are capable of listing in any order. A Priority List produced by the DTA is over and over again a decreasing-time list as shown in fig 2.1.

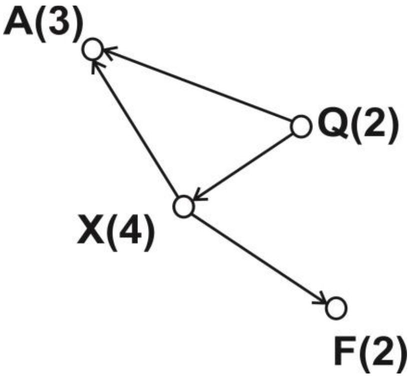

Fig 2.1 decreasing-time Algorithm

One time, it is seen that the precedence relations always overrule the Priority List as soon as there is a conflict involving the two. As a result, for example, at this time the task X cannot in fact be assigned first despite of the fact that it is first on the Priority List from the time when precedence relations insist to facilitate task Q lead task X.

Even if the approach of scheduling says that the longer tasks first are good, it does have a major defect. The DTA pay no attention to any information in the project diagram that shows that one or more tasks ought to be done near the beginning rather than late. For illustration, if one or more tasks by way of long processing times can't commence in anticipation of task X to get finished, at that time passing on task X early will almost certainly result in a shorter finishing time still however assigning task X early go against the DTA.

Critical-Path Algorithm

Formerly, the theory of critical time is known, now we will study about Critical-Path Algorithm. The Critical-Path Algorithm (CPA) is based on an approach comparable to with the aim of Decreasing-Time Algorithm:

It performs the work with high critical times first as well as keeps the jobs with shorter critical times for final. It is seen that, the CPA produce a Priority List by listing the work in declining order of significant times. It is found that work with equal critical times can be listed in any manner. A Priority List created by the CPA is often called a critical-path list.

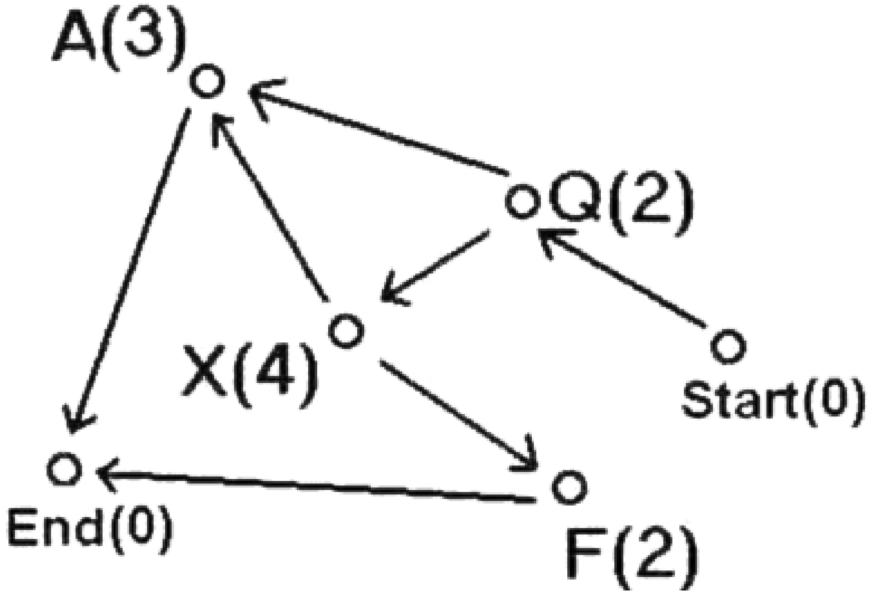

Fig 2.2 Critical-Path Algorithm

The initial step in applying the CPA to a project diagram is to understand the Backflow Algorithm to return all processing times with critical times. Although the Critical-Path Algorithm is usually enhanced as compared to Decreasing-Time Algorithm, neither is guaranteed to produce an optimal schedule. In fact, no efficient scheduling algorithm is presently known that always gives an optimal schedule. However, the Critical-Path Algorithm is the best general-purpose scheduling algorithm currently known.

Check your progress 4

1. Scheduling is :

a. allowing a job to use the processor   c. Both a and b

b. making proper use of processor   d. None of these

2.6 Threads

A thread is a particular sequence stream surrounded by a process. For the reason that threads have a number of properties of processes, they are occasionally called light weight processes. In a process, threads permit multiple implementations of streams. In numerous reverences, threads are accepted way to get better application through parallelism. The CPU switches quickly back as well as forth in the middle of the threads giving false impression that the threads are running in parallel. Like a conventional process i.e., process with one thread, a thread can be in any of several states. Each thread has its individual stack. In view of the fact that thread will usually call different procedures moreover thus a different execution history. This is why thread needs its individual stack. An operating system that has thread facility, the fundamental unit of CPU operation is a thread. A thread has or consists of a program counter (PC), a register set as well as stack space. Threads are not self-governing of one other like processes as a result threads distribute with other threads their code section, data section, OS resources also known as task, such as open files and signals.

Threads are used in designing operating systems because:

- A process with multiple threads makes a great server for example printer server.

- Because threads can share common data, they do not need to use interposes communication.
- Because of the very nature, threads can take advantage of multiprocessors.

Threads are cheap in the intelligence because:

- They only need a stack along with storage for registers as a result, threads are cheap to create.
- Threads use very small resources of an operating system in which they are working. That is, threads do not require new address space, global data, program code or operating system resources.
- Context switching is fast as soon as working with threads. The reason is that we only have to save and/or restore PC, SP and registers.

As shown in Figure 2.3, multi-threaded applications contain multiple threads contained by single process, all having their individual program counter, stack and set of registers, other than sharing common code, data as well as certain structures such as open files

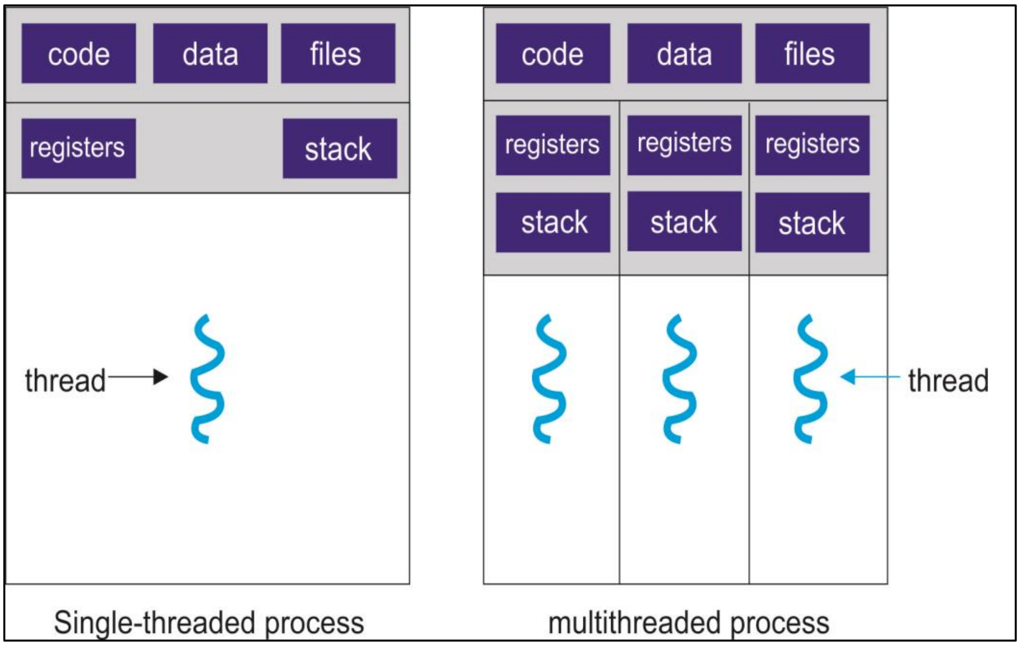

Fig 2.3 Single and multi-threaded

Architecture

Threads are extremely useful in modern programming at any time the process has multiple tasks to carry out independently of the others.

This is on the whole true when one of the tasks possibly will block, furthermore it is required to allow the other tasks to proceed with no blocking.

For instance in a word processor, a surroundings thread may ensure spelling as well as grammar while a centre thread processes user input, while however a third thread loads images from the hard drive, as well as a fourth does periodic automatic backups of the file being condensed.

An additional instance is a web server - Multiple threads permit for multiple requests to be fulfilled simultaneously, with no service requests sequentially or to fork off separate processes for each incoming request.

Benefits

There are four major categories of benefits to multi-threading:

Responsiveness - One thread may give rapid reply at the same time other threads are blocked-up or slow down doing serious calculations.

Resource sharing - By logic, threads contribute to common code, data, as well as other resources, which allows numerous tasks to be performed at the same time in a single address space.

Economy - Creating as well as managing threads is a lot faster than performing the same tasks for processes.

Scalability, i.e. Utilization of multiprocessor architectures - A single threaded process be able to only run on one CPU, no issue how many may be accessible, while the execution of a multi-threaded application might be split in the middle of available processors.

Check your progress 5

1. A process can be_________.

a. single threaded c. both (a) and (b)

b. multithreaded d. none of the mentioned

2. Which of the following is not a valid state of a thread?

a. running c. ready

b. parsing d. blocked

2.7 Let Us Sum Up

In this unit we have learned:

- That a virtual CPU also called as CPU is called as a virtual processor.
- In this, an interrupt is a signal from equipment affixed to a computer.
- We see that thread is the smallest unit of processing that can be performed in an operating system.

2.8 Answers for Check Your Progress

Check your progress 1

Answers: (1-a), (2-b)

Check your progress 2

Answers: (1-b)

Check your progress 3

Answers: (1-c), (2-d)

Check your progress 4

Answers: (1-c)

Check your progress 5

Answers: (1-c), (2-b)

2.9 Glossary

1. **Virtual reality -** Virtual reality is an artificial environment that is created with software and presented to the user in such a way that the user suspends belief and accepts it as a real environment.

2. **VMware Storage Policy-Based Management -** Storage Policy-Based Management is a feature that allows for automatic provisioning of virtual machines in a VMware environment.

3. **VMware Platform Services Controller (PSC)** - VMware Platform Services Controller (PSC) is a new service in vSphere 6 that handles the infrastructure security functions.

4. **Virtualization -** Terms related to virtualization, including definitions about virtualization technologies and words and phrases about server virtualization, desktop virtualization and storage virtualization.

2.10 Assignment

Write detail on Page replacement algorithms.

2.11 Activities

Explain Paging address Translation by direct mapping.

2.12 Case Study

Write the different types of thread mechanism.

2.13 Further Reading

1. The Operating system by Andrew Tannenbaum.
2. Operating System by Mach.