5.3 Explores how an operating system manages processes in computers.

Time: 6 periods

Learning Outcomes

- Explains processes
- Lists the operating system tasks when a process is created
- Lists the types of processes
- Lists the process states
- Explains process termination
- Distinguishes a process and a program
- Explains process states using the seven state process transition diagrams
- Describes process schedulers and scheduling policies
- Compares long, short and medium term schedulers
- Describes multi programming and its needs
- Describes time sharing systems
- Compares multi programming vs. time sharing systems
- Defines context switch
- Briefly explains turnaround time, response time, throughput time and waiting time
- Briefly explains the process control block and lists its contents

What is Process?

A program in execution or a program that can be assigned to and executed by a processor. Process is not a program. A program may have many processes.

A process consists of,

- Executable codes
- Data needed for execution
- Execution context (status of execution such as address in PC, priorities, waiting for I/O or not, …)

Interrupt Handling

What is an interrupt?

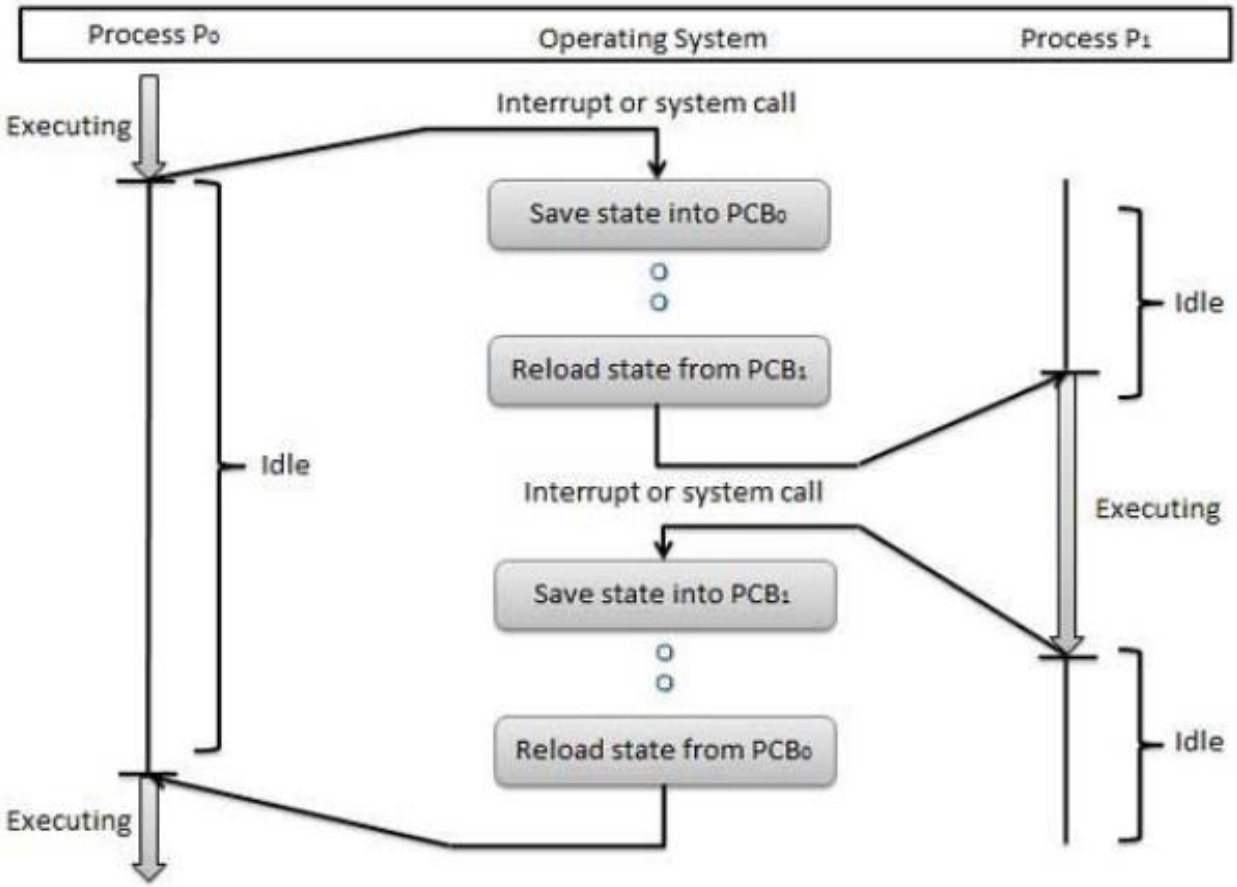
Interrupt is an event that alters the sequence of execution of processes.

How to handle interrupts by OS

Generally I/O devices are much slower than processor. Therefore, after an I/O call, processor has to sit idle till the I/O device completes its event. Thus, OS saves execution context of current process and switches processor to execute some other process. When I/O event completed, I/O device interrupts OS. Operating System then retrieves the execution context of suspended process and switches the processor to resume its execution from where it was suspended.

Context Switching

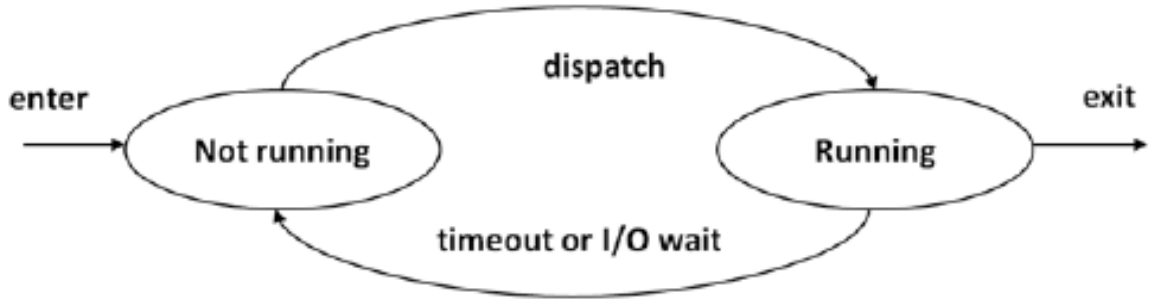
- A context switch is the mechanism to store and restore the state or context of a CPU in Process Control block so that a process execution can be resumed from the same point at a later time.
- Using this technique a context switcher enables multiple processes to share a single CPU. Context switching is an essential part of a multitasking operating system features.
- When the scheduler switches the CPU from executing one process to execute another, the context switcher saves the content of all processor registers for the process being removed from the CPU, in its process control block.
- Context switch time is pure overhead.
- Context switching can significantly affect performance as modern computers have a lot of general and status registers to be saved.



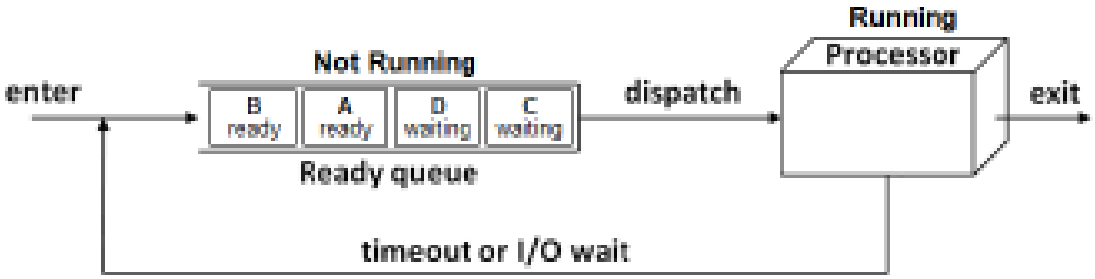
Execution of Processes

In multi-programmed time sharing systems, memory holds multiple processes (programs in execution).Operating System switches processor from one process to another after a certain time quantum (context switching).When current process wants to do I/O, Operating System switches processor to execute another process in memory. Operating System uses a process model to manage the execution of processes.

Two state process model



New processes enter into **Not Running** state. After the expiry of time quantum or process wants to do I/O, Operating System interrupts the process in **Running** state and transfers it to **Not Running** state. Operating System then dispatches another process in **Not Running** state to **Running** state for execution. Complete/aborted processes exit from **Running** state.



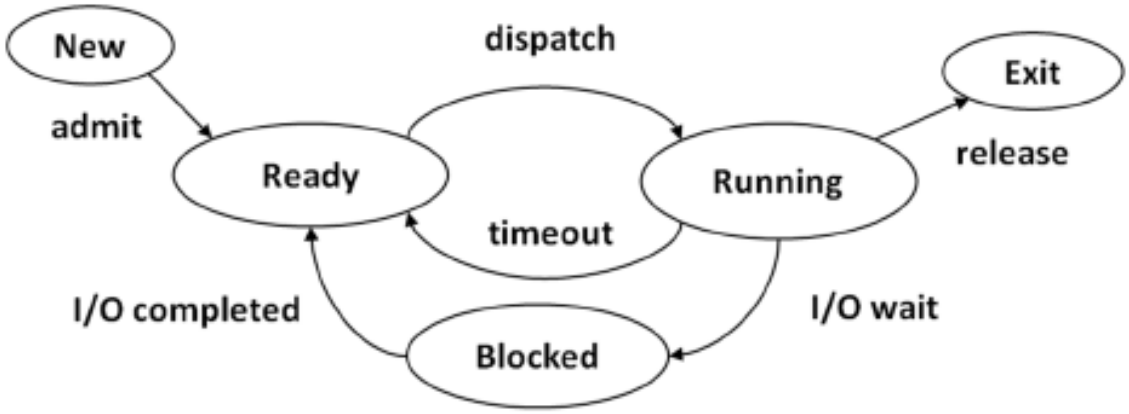
- **Problem** of Two status process model

Some processes in **Not Running** state may be ready to run, but can be blocked by processes waiting for some I/O event completion.

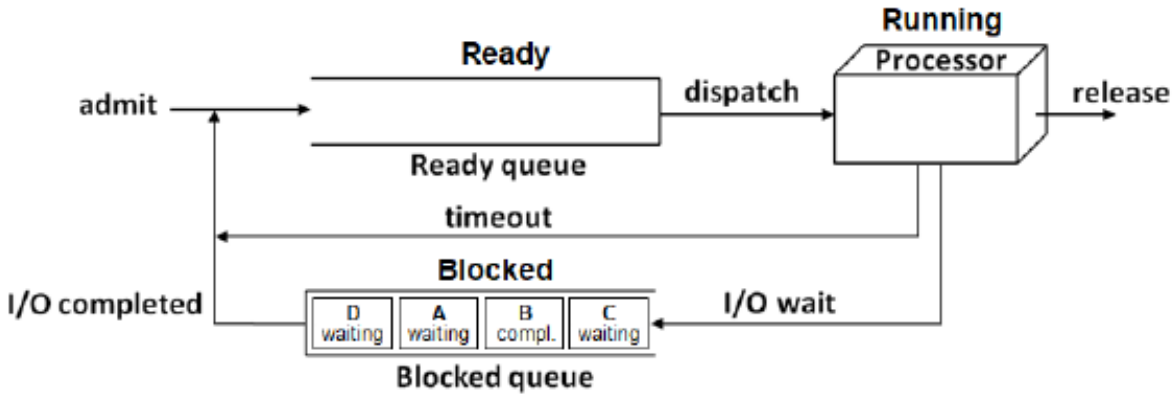
- **Solution**

Split the **Not Running** state into **Ready** and **Blocked** states.

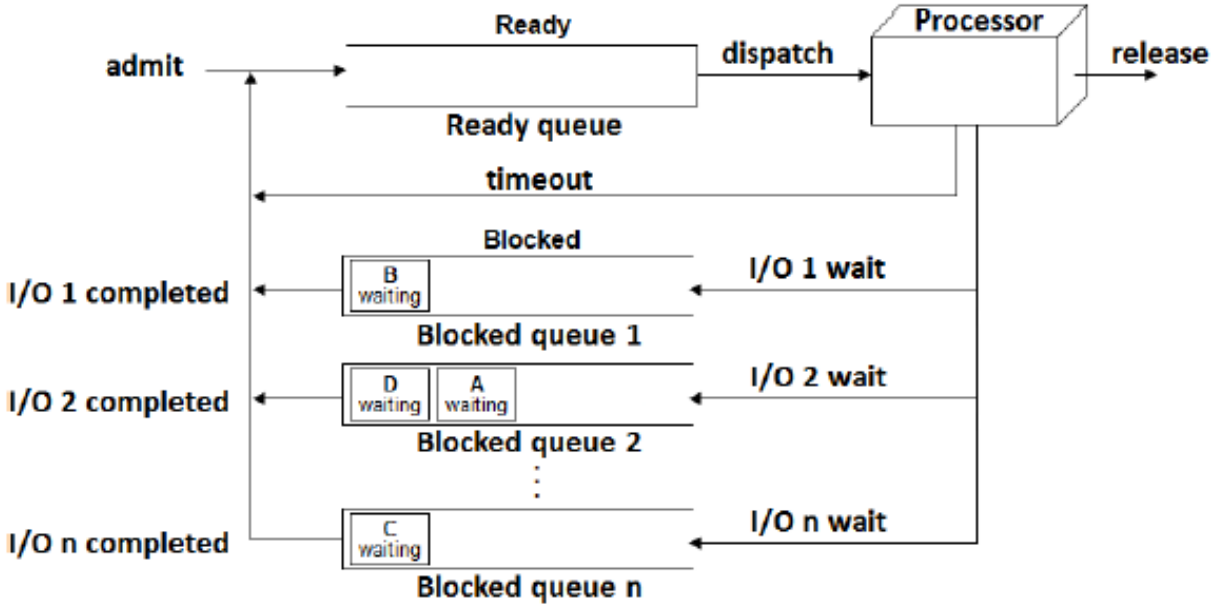
Five State Process Model



- **New** – just created, not in memory (on Dard disk)
- **Ready** – in memory, ready to run
- **Running** – being executed
- **Blocked** – in memory, not ready to run, waiting for I/O
- **Exit** – execution completed or aborted

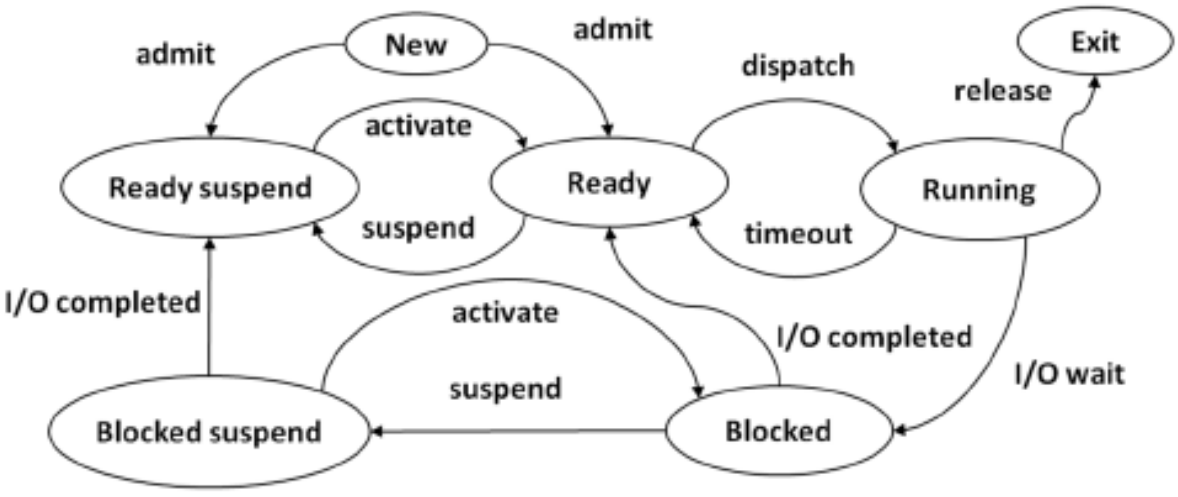


- **Problem** - Some processes in **Blocked** state may be blocked even after their I/O events are completed.
- **Solution -** Split the **Blocked** state to handle different I/O events.



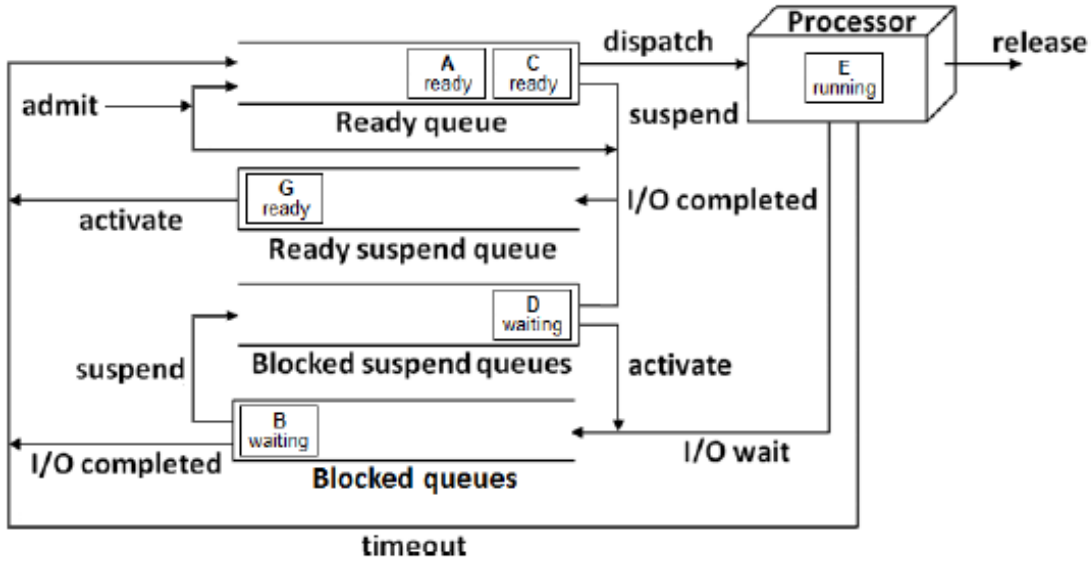
- **Problem** – If all processes in memory are blocked for I/O and memory is full, processor would become idle.
- **Solution** – Move some processes from memory to disk.

Seven State Process Model



- **Blocked suspend** – on disk, waiting for I/O
- **Ready suspend** – on disk, ready to run when loaded into memory

When no process is ready to run and memory is full, one or more processes are swapped out and one or more new or suspended processes are swapped in.



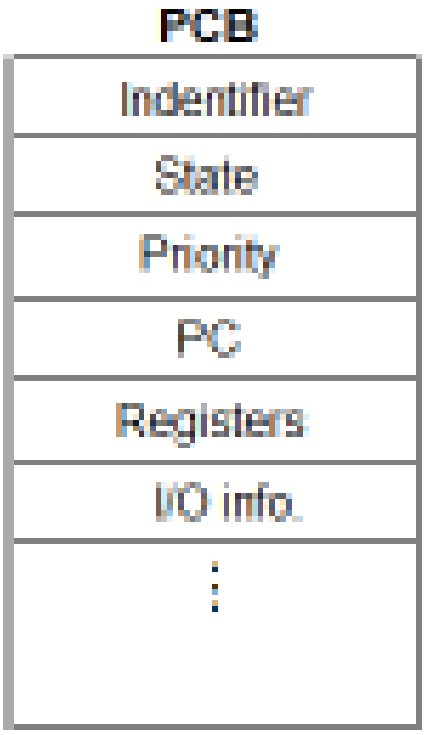
Note the **Blocked** and **Blocked suspend** states are implemented with multiple queues to handle processes waiting for different I/O events.

Process Control Block

A data structure created and maintained by OS to manage a process. It holds information related to the process. When suspending the execution of a process, OS saves **execution context** of the process in PCB and later uses it to resume the execution from where it was suspended. When process is completed or aborted its execution, OS deleted its PCB. Information stored in PCB is depends on OS.

Information Stored in PCB

- **Identifier** – unique identifier for OS to distinguish process from other processes
- **State** – state of the process
- **Priority** – priority relative to other processes
- **PC** – address of the next instruction of the process to be executed
- **Registers** – CPU register values associated with the process
- **I/O info** – I/O devices assigned to process, any outstanding I/O requests, etc.



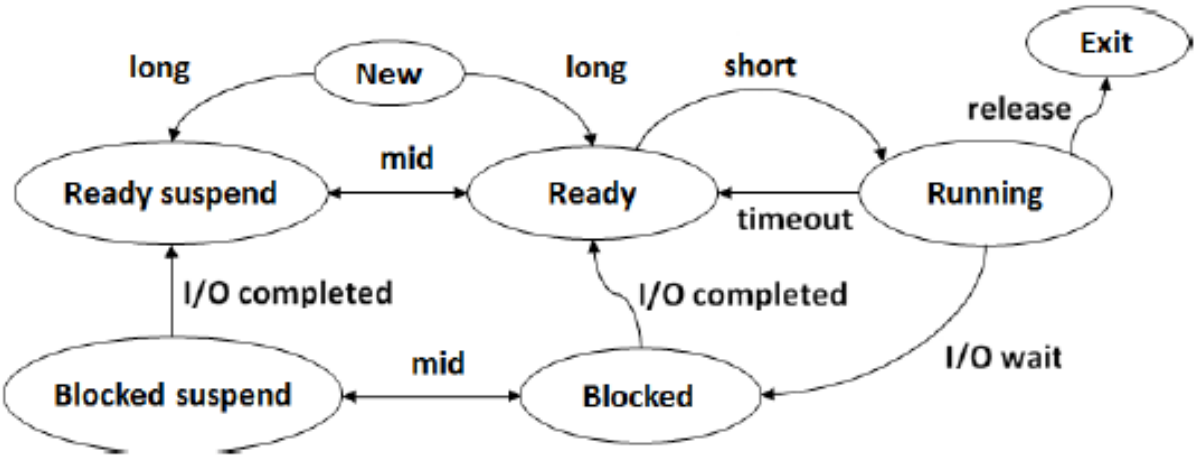
Types of Processes

- **Processor bound processor** – Spend more time for using processor than doing I/O
- **I/O bound processes** – Spend more time for doing I/O than using processor
- **Preemptive processes** – can be interrupted, suspended the execution and resumed later
- **Non – preemptive processes** – Cannot be interrupted and continue until process completes its execution or blocked by itself

Process Scheduling

- **Long term scheduling –** schedules admission of processes, decides degree of multiprogramming
- **Mid-term scheduling** - schedules swapping function

- **Short term scheduling** – schedules switching of processes between ready and running state



Scheduler Comparison

Long Term Scheduler	Short Term Scheduler	Medium Term Scheduler
Job Scheduler	CPU scheduler	Processes swapping scheduler
Selects processes from a pool and loads them into the memory for execution	Selects those processes which are ready to execute for dispatching	Swapped out/Re-introduces the processes into memory and execution can be continued.
Controls the degree of multiprogramming	Provides lesser control over the degree of multiprogramming	Controls the degree of multiprogramming
Speed is lesser than short term scheduler	Speed is fastest among other two	Speed is in between (short and long term schedulers)

Assignment of processes to processor for execution in a way that optimizes (educes/increases) one or more objectives.

- **Response time** – time between submission of a request and receipt of response
- **Throughput** – number of processes completed per unit time

- **Turnaround time** – average time from submission of a process to its completion
- **Waiting time** – time a process spends in ready state

Impossible to optimize all of above at the same time

References

Teachers Guide – 2017

Chapter 3 and 9, W.Stallings, Operating System: Internals and Design Principles, 9th Edition, Pearson, 2017