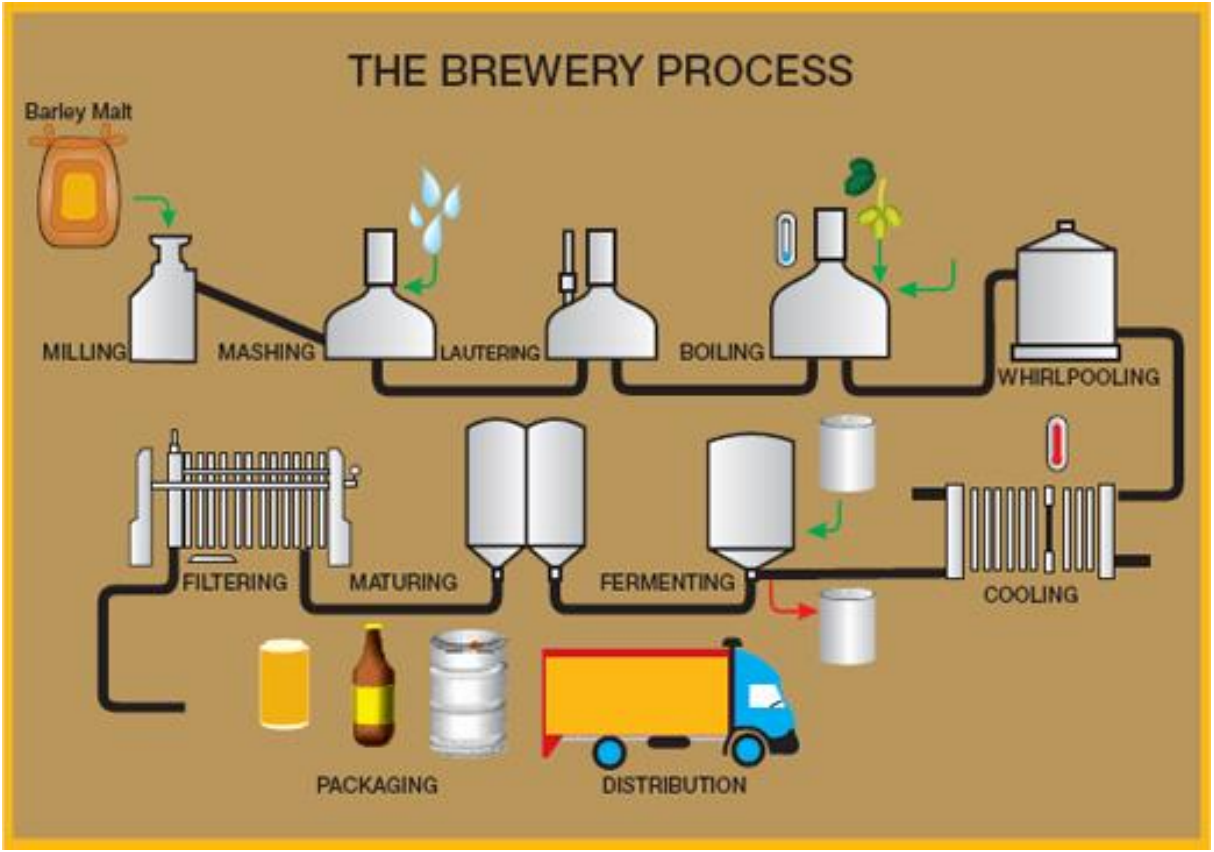
Chapter 3 – Process Description and Control
Lecture 3

Roadmap

- **How are processes represented and controlled by the OS?**
- *Process states* which characterize the behaviour of processes.
- *Data structures* used to manage processes.
- Ways in which the OS uses these data structures to control process execution.

What are the fundamental tasks for an operating system?

Process management



Process

Fundamental to the structure of OS's

A *process* is:

- A program in execution
- An instance of a running program
- The entity that can be assigned to and executed on a processor
- A single sequential thread of execution, a current state, and an associated set of system resources.

Process elements

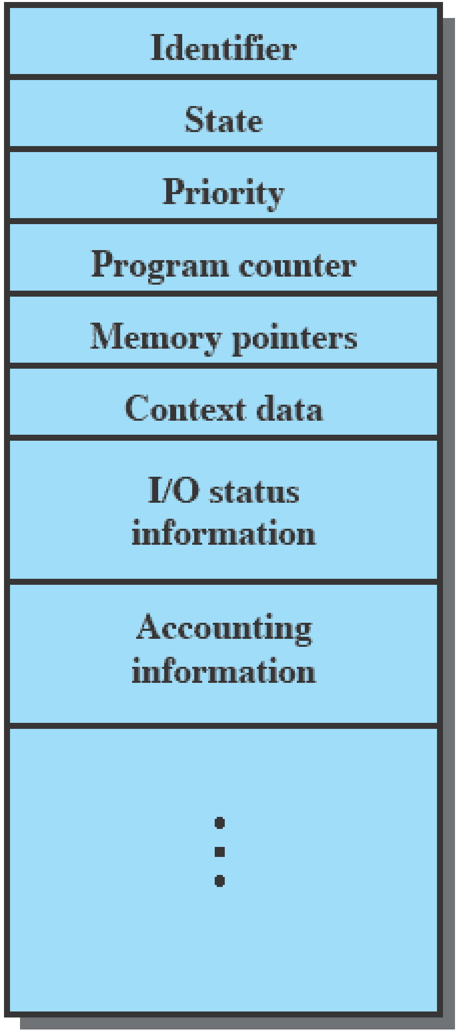
Program code

Set of data

Status info

Process Control Block

- Contains the process elements
- It is possible to interrupt a running process and later resume execution as if the interruption had not occurred
- Created and manage by the operating system
- Key tool that allows support for multiple processes



Identifier

State

Priority

Program counter

Memory pointers

Context data

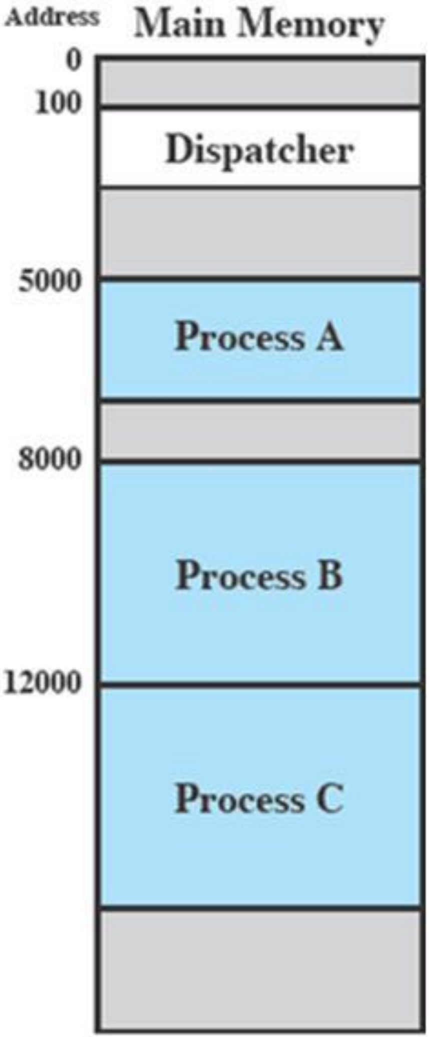
I/O status information

Accounting information

⋮

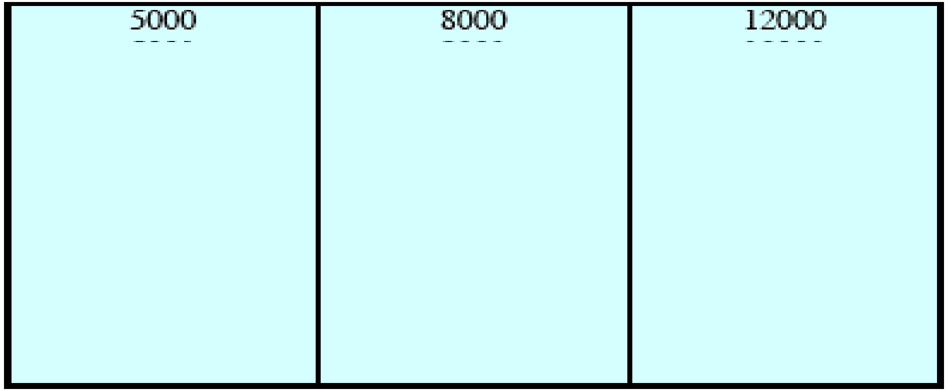
Process Execution

- Consider three processes being executed
- All are in memory (plus the dispatcher)
- Lets ignore virtual memory for this.



Trace from the *processes* point of view:

- Each process runs to completion

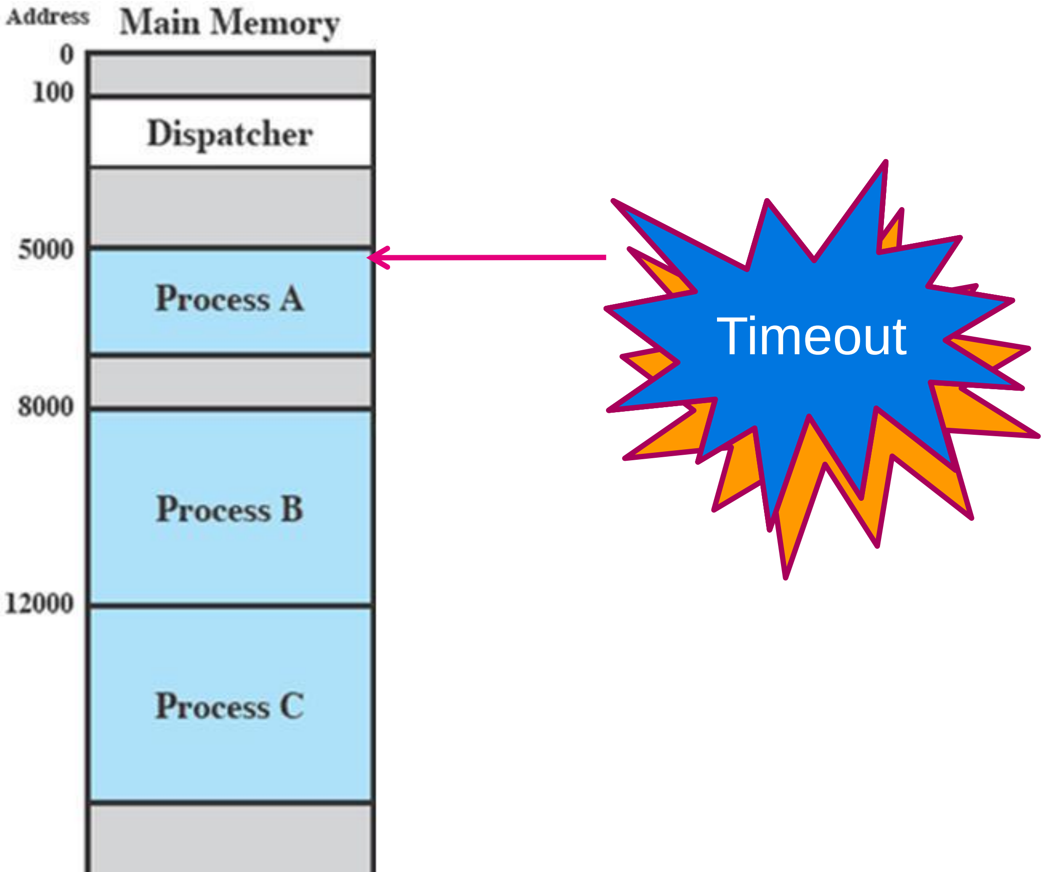


(a) Trace of Process A	(b) Trace of Process B	(c) Trace of Process C
5000	8000	12000

5000 = Starting address of program of Process A
8000 = Starting address of program of Process B
12000 = Starting address of program of Process C

Figure 3.3 Traces of Processes of Figure 3.2

Trace from *Processors* point of view



```
 1    5000
 2    5001
 3    5002
 4    5003
 5    5004
 6    5005
------------------- Timeout
 7    100
 8    101
 9    102
10    103
11    104
12    105
13    8000
14    8001
15    8002
16    8003
----------------- I/O Request
17    100
18    101
19    102
20    103
21    104
22    105
23    12000
24    12001
25    12002
26    12003
```

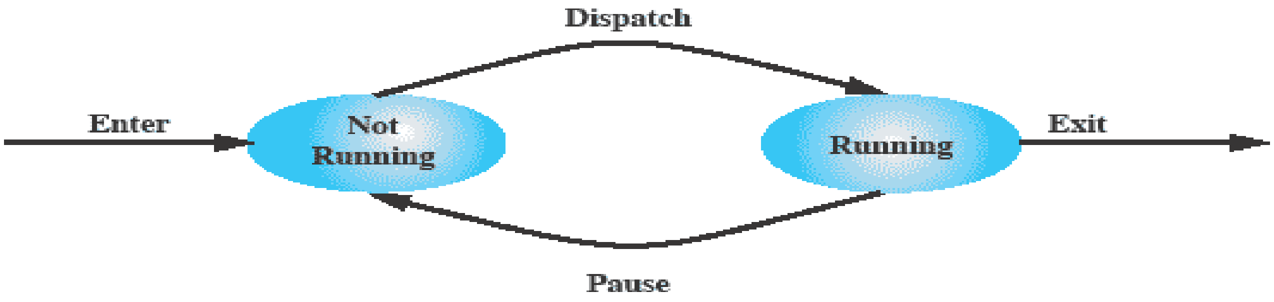
```
27    12004
28    12005
------------------- Timeout
29    100
30    101
31    102
32    103
33    104
34    105
35    5006
36    5007
37    5008
38    5009
39    5010
40    5011
------------------- Timeout
41    100
42    101
43    102
44    103
45    104
46    105
47    12006
48    12007
49    12008
50    12009
51    12010
52    12011
------------------- Timeout
```

Roadmap

- How are processes represented and controlled by the OS?
- ***Process states* which characterize the behaviour of processes**.
- *Data structures* used to manage processes.
- Ways in which the OS uses these data structures to control process execution.

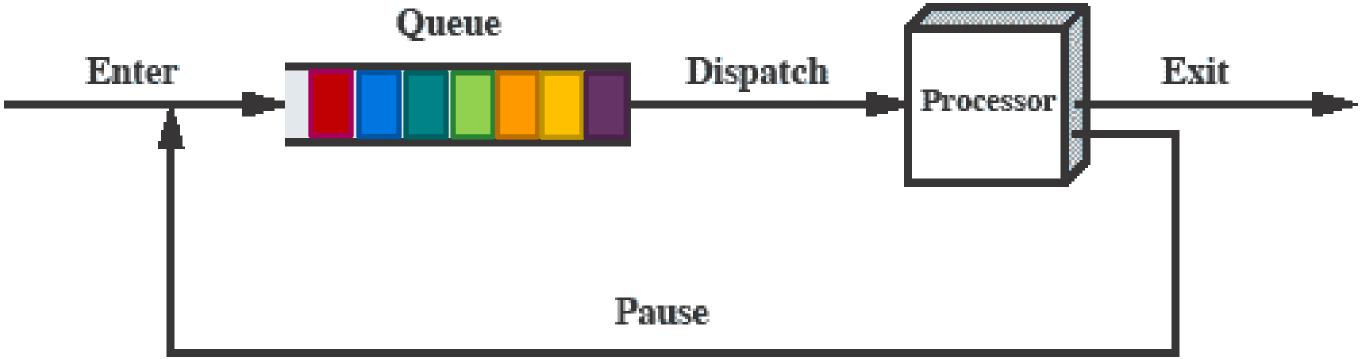
Two-State Process Model

- Process may be in one of two states
  - Running
  - Not-running



(a) State transition diagram

Queuing Diagram



(b) Queuing diagram

Etc … processes moved by the dispatcher of the OS to the CPU then back to the queue until the task is competed

Process Birth and Death

Reasons for Creation	Termination
New batch job	Normal Completion
Interactive Login	Memory unavailable
Created by OS to provide a service	Protection error
Spawned by existing process	Operator or OS Intervention

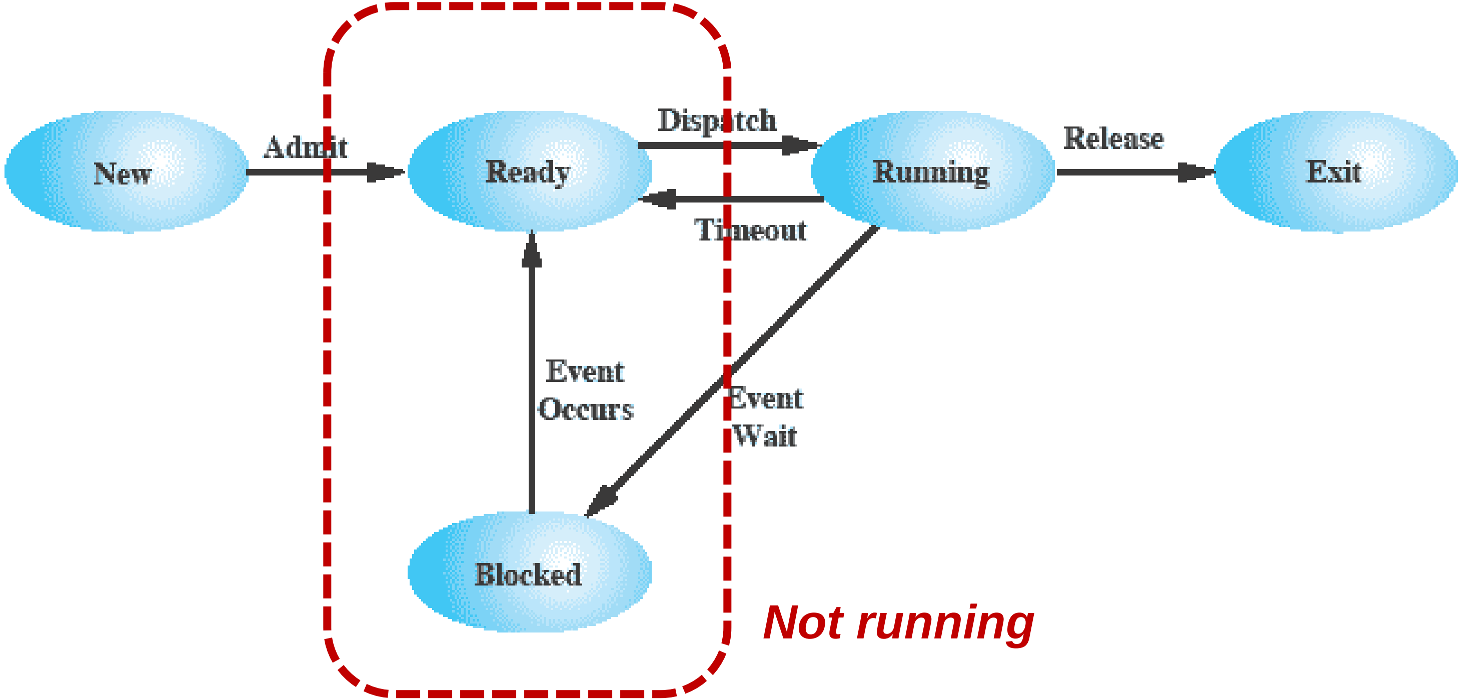
See tables 3.1 and 3.2 for more

Process Creation

What needs to be done?

Who creates a process?
And termination?

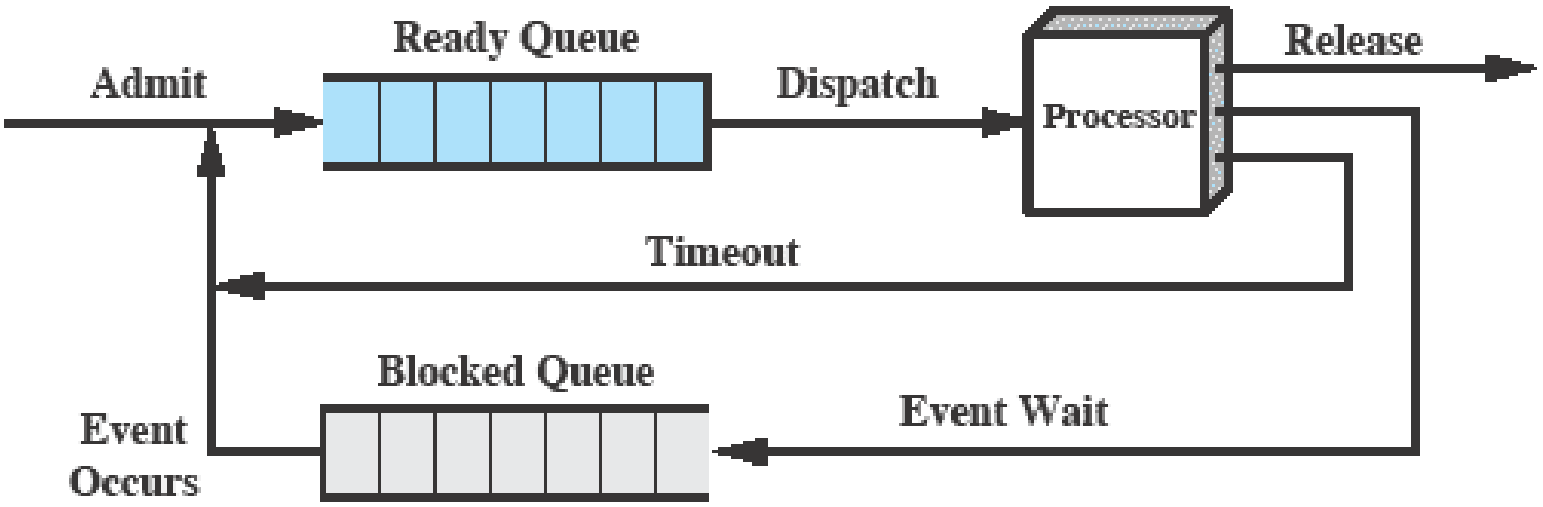
Five-State Process Model



Not running

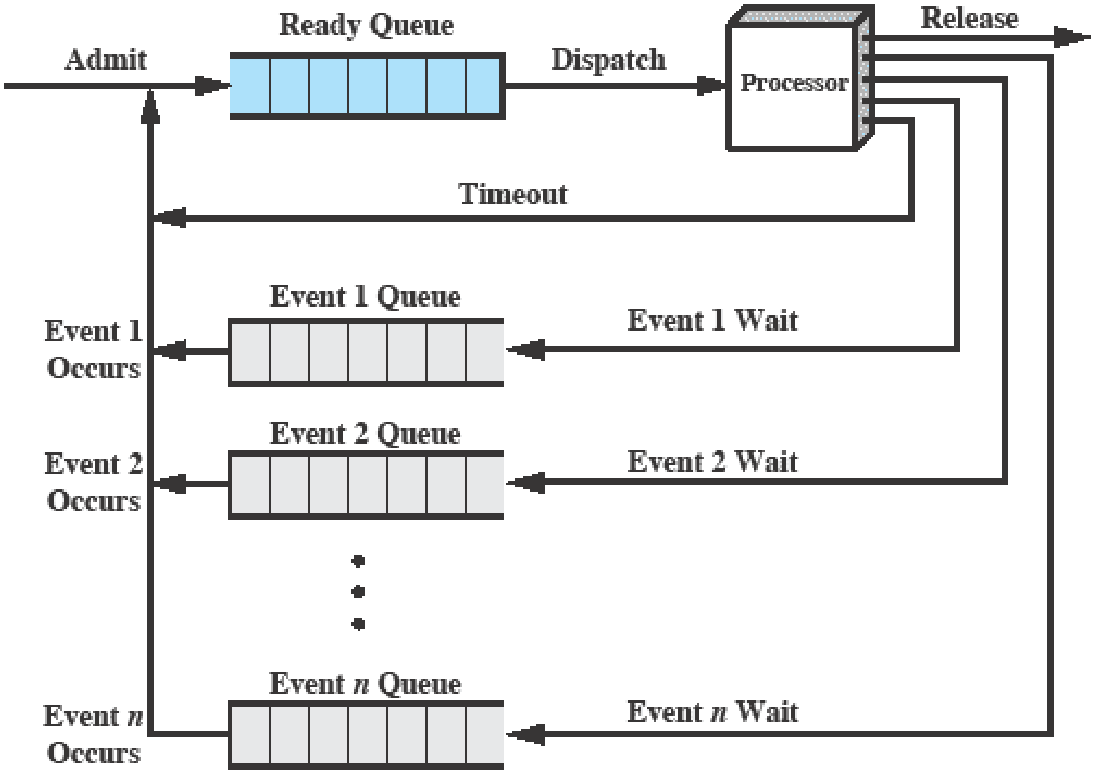
Figure 3.6 Five-State Process Model

Using Two Queues



(a) Single blocked queue

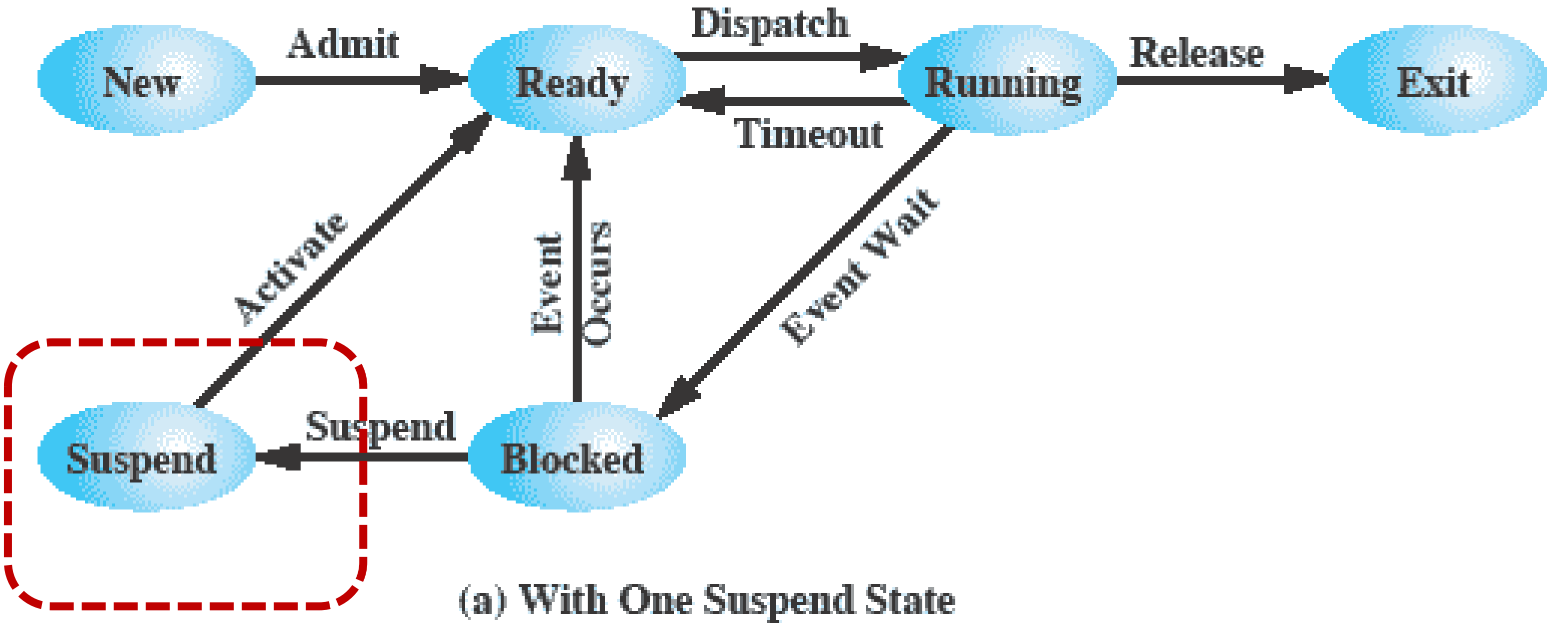
Multiple Blocked Queues



(b) Multiple blocked queues

What (should) happen(s) when there are too many processes?

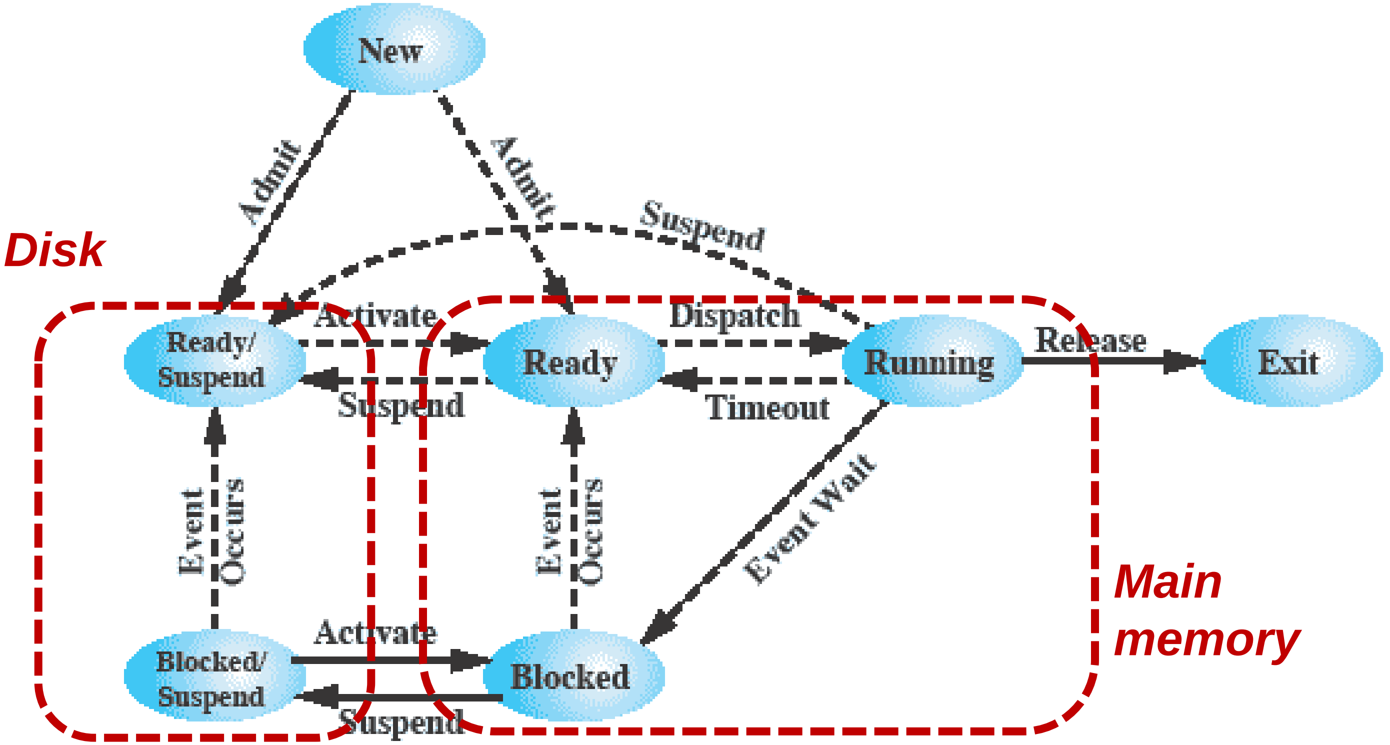
One Suspend State



(a) With One Suspend State

Disk

Two Suspend States

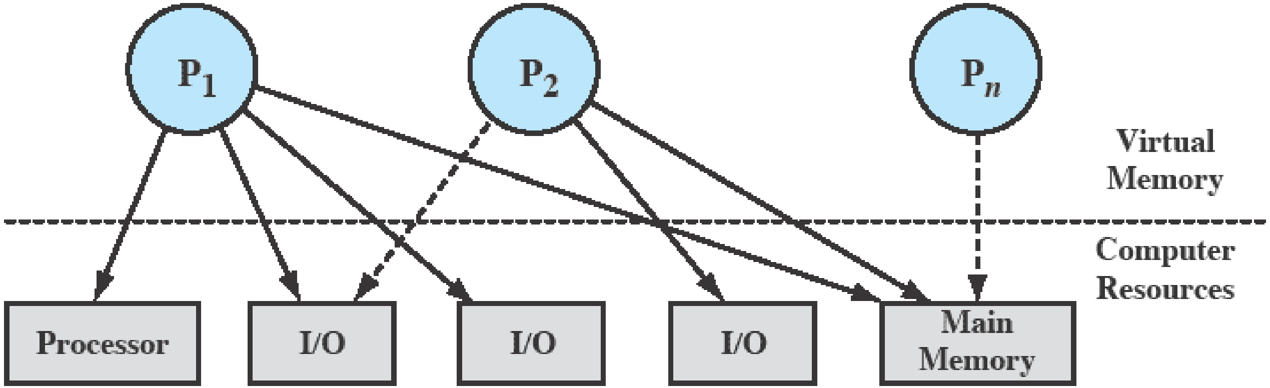


(b) With Two Suspend States

Roadmap

- How are processes represented and controlled by the OS?
- *Process states* which characterize the behaviour of processes.
- ***Data structures* used to manage processes.**
- Ways in which the OS uses these data structures to control process execution.

Processes and Resources



Tables are needed to manage status and resources

Figure 3.10 Processes and Resources (resource allocation at one snapshot in time)

OS Control Tables

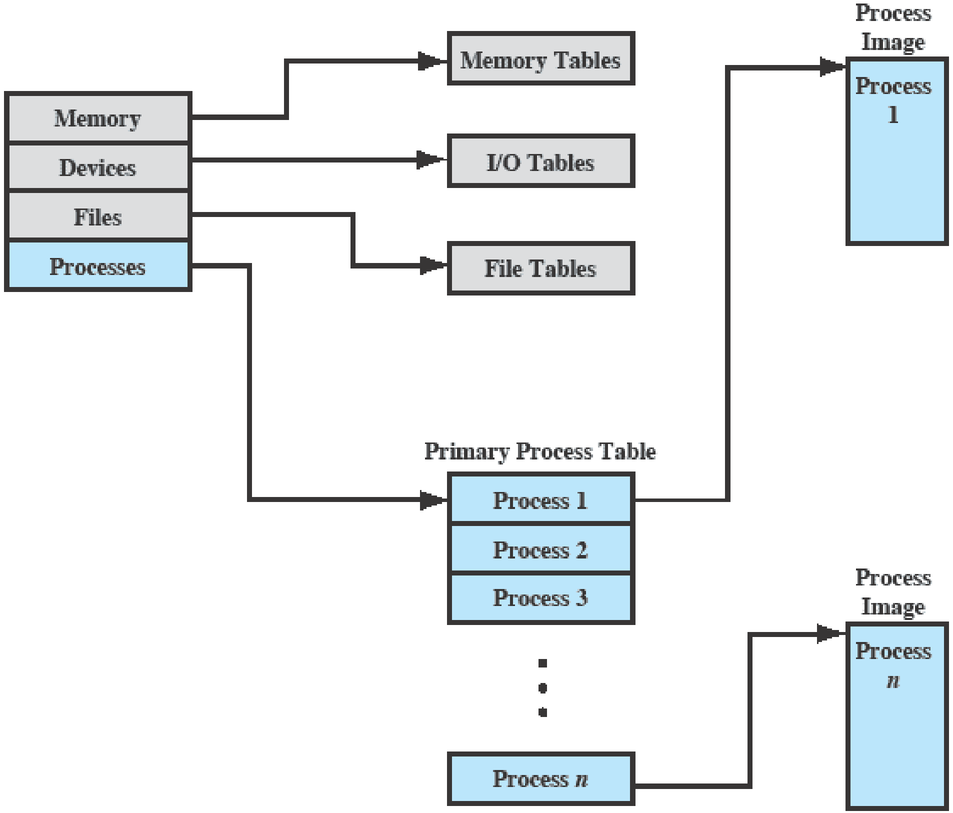
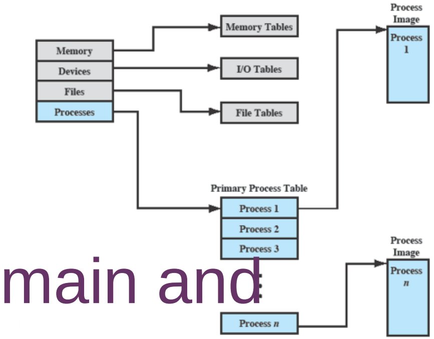


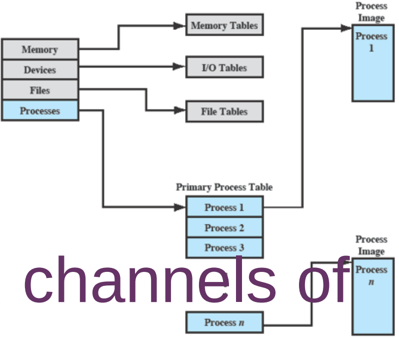
Figure 3.11 General Structure of Operating System Control Tables

Memory Tables



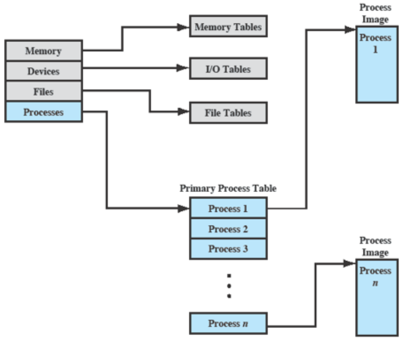
- Memory tables are used to keep track of both main and secondary memory.
- Must include this information:
  - Allocation of main memory to processes
  - Allocation of secondary memory to processes
  - Protection attributes for access to shared memory regions
  - Information needed to manage virtual memory

I/O Tables



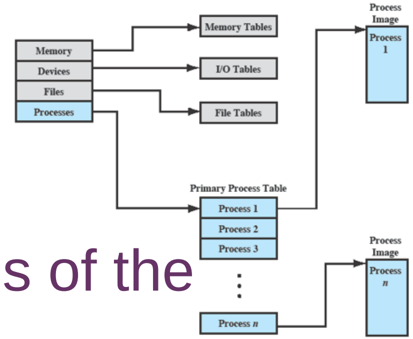
- Used by the OS to manage the I/O devices and channels of the computer.
- The OS needs to know
  - Whether the I/O device is available or assigned
  - The status of I/O operation
  - The location in main memory being used as the source or destination of the I/O transfer

File Tables



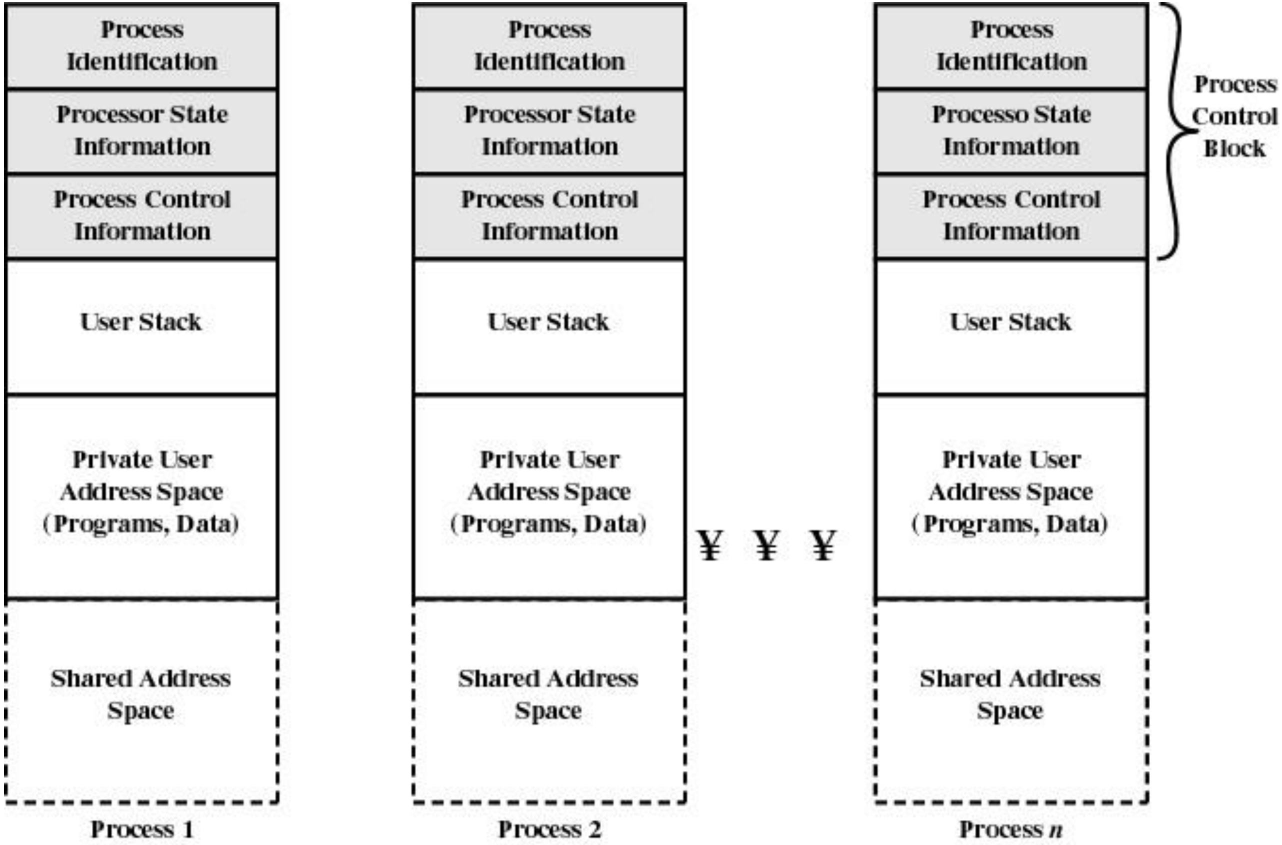
- These tables provide information about:
  - Existence of files
  - Location on secondary memory
  - Current Status
  - other attributes.
- Sometimes this information is maintained by a file management system

Process Tables



- To manage processes the OS needs to know details of the processes
  - Current state
  - Process ID
  - Location in memory
  - etc
- Process control block
  - ***Process image*** is the collection of program. Data, stack, and attributes

Process Image

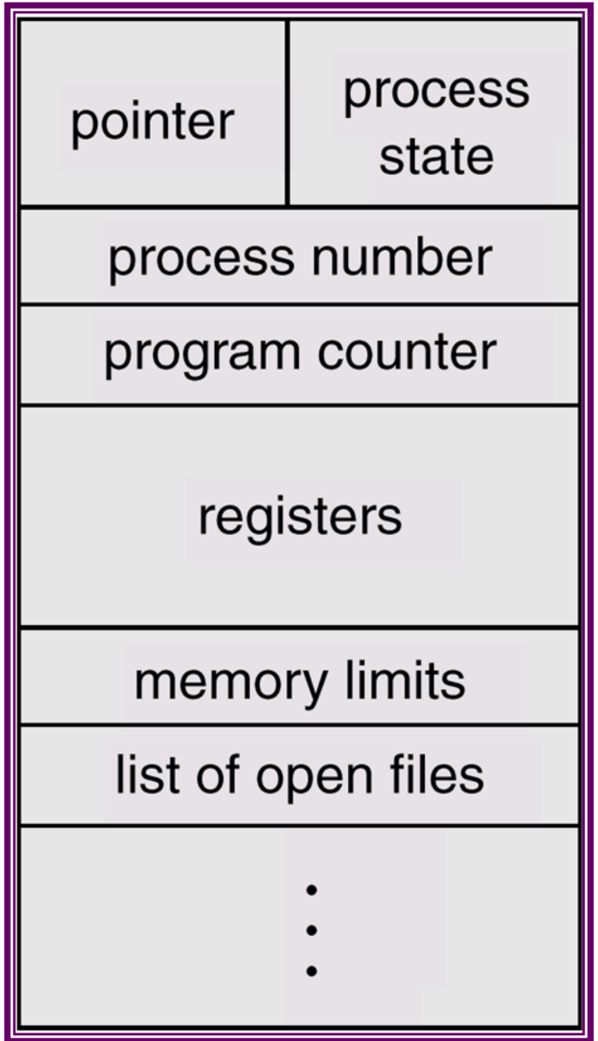

Process Identification
Processor State Information
Process Control Information
User Stack
Private User Address Space (Programs, Data)
Shared Address Space
Process 1
Process Identification
Processor State Information
Process Control Information
User Stack
Private User Address Space (Programs, Data)
Shared Address Space
Process 2
¥ ¥ ¥
Process Identification
Processo State Information
Process Control Information
User Stack
Private User Address Space (Programs, Data)
Shared Address Space
Process *n*
Process Control Block

Process Control Block

Process Identification

Process status

Process Control information



pointer

process state

process number

program counter

registers

memory limits

list of open files

•
•
•

Role of the Process Control Block

- The most important data structure in an OS
  - It defines the state of the OS
- Process Control Block requires protection
  - A faulty routine could cause damage to the block destroying the OS's ability to manage the process
  - Any design change to the block could affect many modules of the OS

Roadmap

- How are processes represented and controlled by the OS?
- *Process states* which characterize the behaviour of processes.
- *Data structures* used to manage processes.
- **Ways in which the OS uses these data structures to control process execution.**

Process Creation

- Once the OS decides to create a new process it:
  - Assigns a unique process identifier
  - Allocates space for the process
  - Initializes process control block
  - Sets up appropriate linkages such as scheduling queues
  - Creates or expand other data structures *(e.g. accounting)*

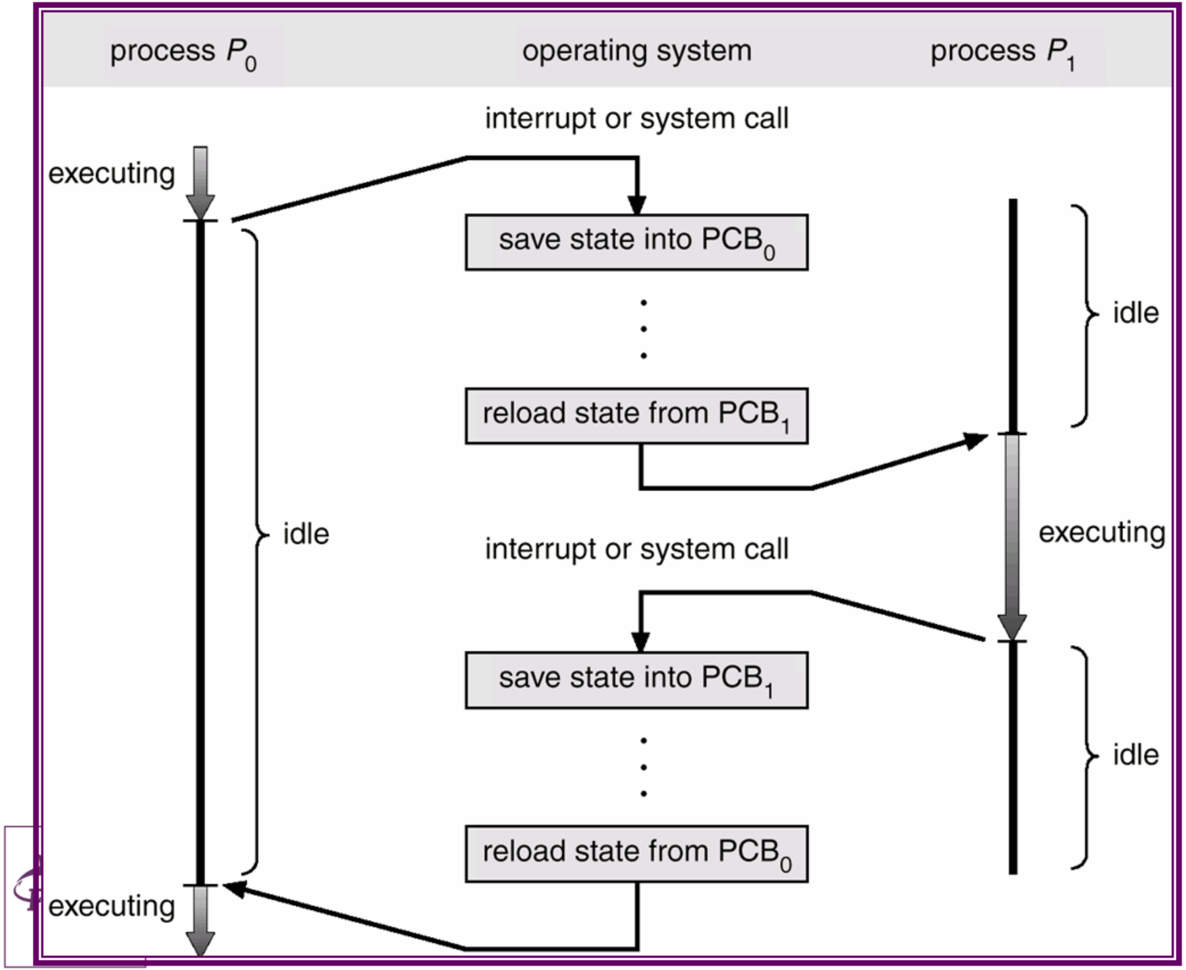
Switching Processes - *When to switch processes*

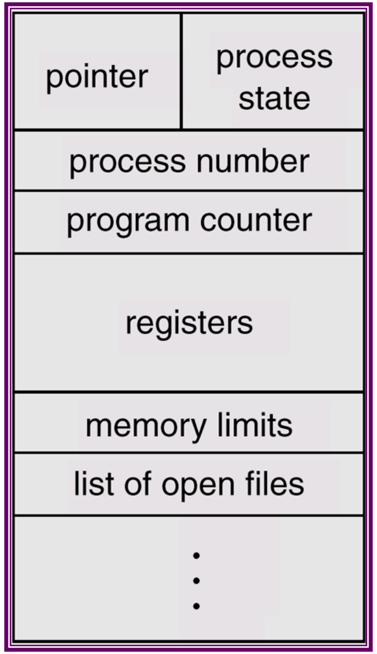
A process switch may occur any time that the OS has gained control from the currently running process. Possible events giving OS control are:

Mechanism	Cause	Use
Interrupt	External to the execution of the current instruction	Reaction to an asynchronous external event
Trap	Associated with the execution of the current instruction	Handling of an error or an exception condition
Supervisor call	Explicit request	Call to an operating system function

Table 3.8

PCB





pointer	process state
process number	
program counter	
registers	
memory limits	
list of open files	
⋮	

Change of Process State …

The steps in a process switch are:

1. Save context of processor including program counter and other registers
2. Update the process control block of the process that is currently in the Running *state*
3. Move process control block to appropriate queue – ready; blocked; ready/suspend

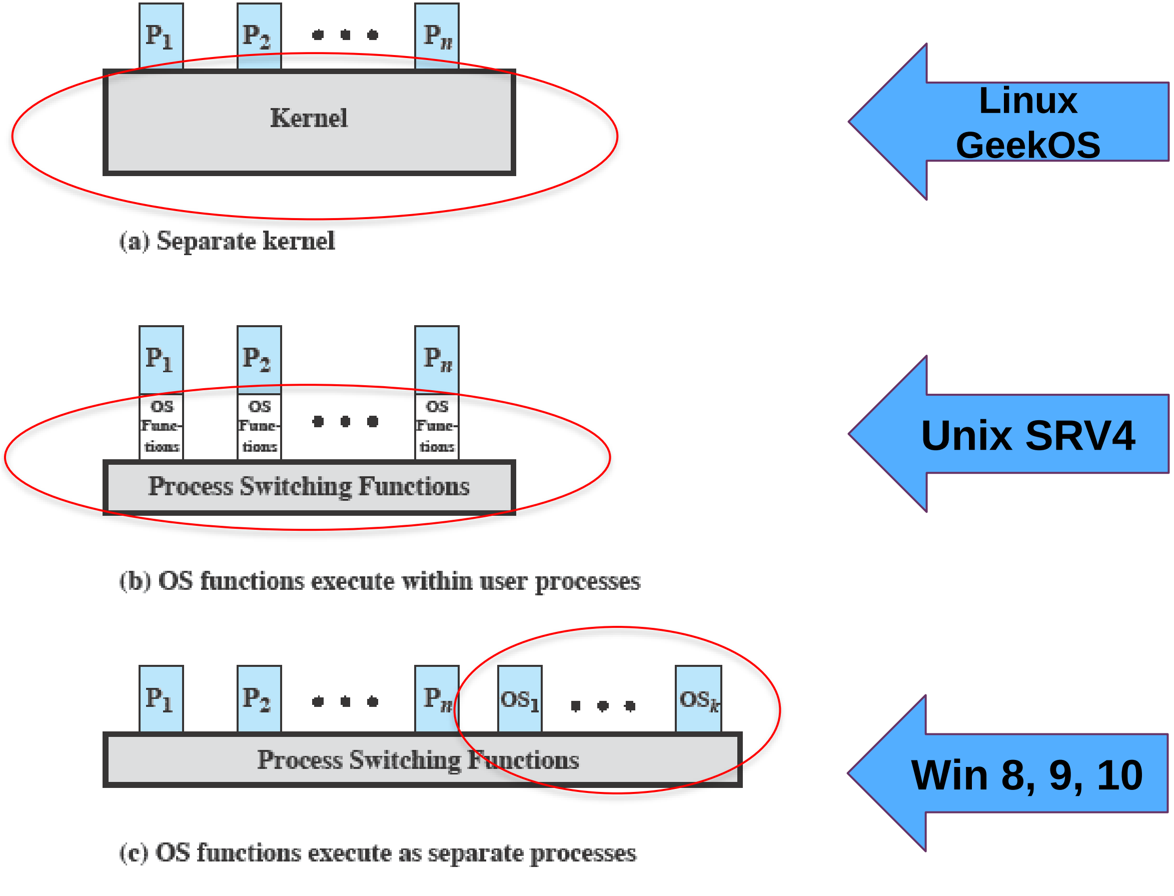
Change of Process State cont…

4. Select another process for execution
5. Update the process control block of the process selected
6. Update memory-management data structures
7. Restore context of the selected process

Is the OS a Process?

- If the OS is just a collection of programs and if it is executed by the processor just like any other program, is the OS a process?
- If so, how is it controlled?
  - Who (what) controls it?

Execution of the Operating System



(a) Separate kernel

Linux
GeekOS

(b) OS functions execute within user processes

Unix SRV4

(c) OS functions execute as separate processes

Win 8, 9, 10

Figure 3.15 Relationship Between Operating System and User Processes

Random selection
&
Practical assignment explanation

Questions?