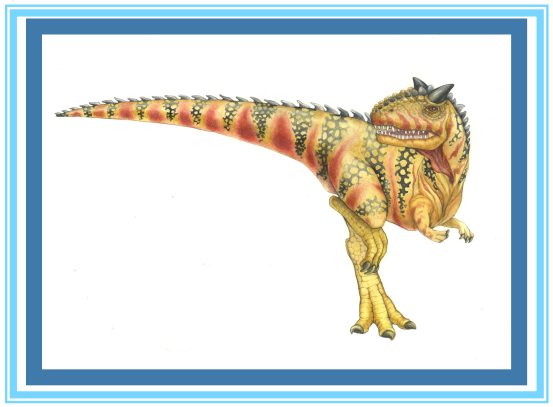
Chapter 3: Processes

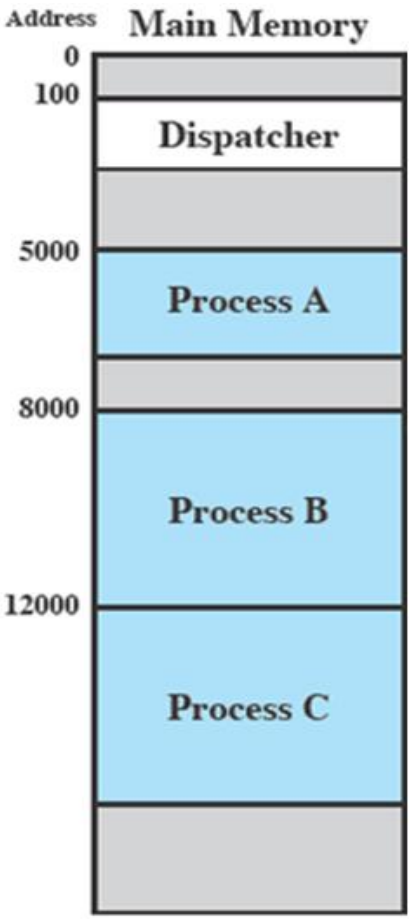


Silberschatz, Galvin and Gagne ©2009

Multi-programming, -tasking, -threading

- **Multiprogramming**
  - Run multiple programs with timesharing
  - Run each program in turn until timeout or I/O event
- **Multitasking**
  - Multiprogramming with frequent job switching
  - CPU scheduling
- **Multithreading**
  - A process with multiple threads

Process Execution with Multiprogramming



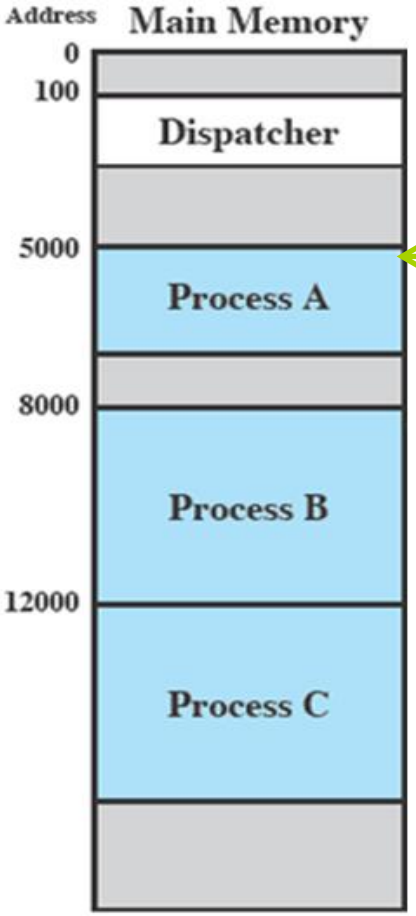
5000	8000	12000
5001	8001	12001
5002	8002	12002
5003	8003	12003
5004		12004
5005		12005
5006		12006
5007		12007
5008		12008
5009		12009
5010		12010
5011		12011

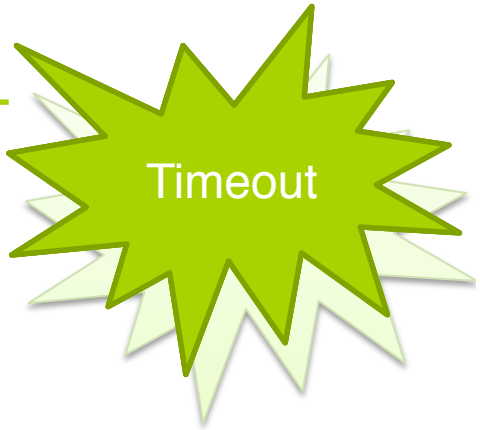
(a) Trace of Process A **(b) Trace of Process B** **(c) Trace of Process C**

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

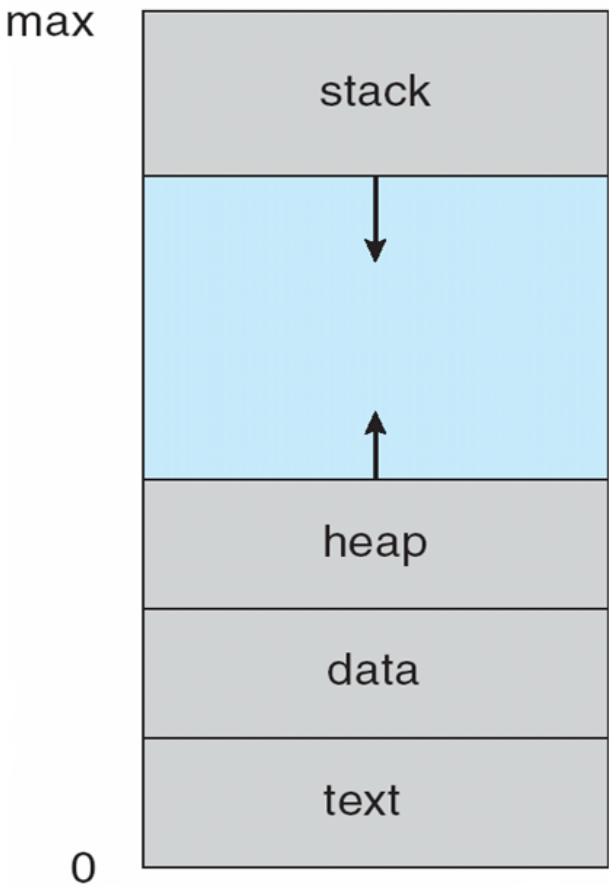
100 = Starting address of dispatcher program

Shaded areas indicate execution of dispatcher process;
first and third columns count instruction cycles;
second and fourth columns show address of instruction being executed

Figure 3.4 Combined Trace of Processes of Figure 3.2

Process Concept

- Process a concept
  - A program in execution (memory)
  - *job* = *process*
  - Process = code + data
- Process a data structure
  - Information about a process
  - Allows context switching
- Context switching
  - Switch between processes

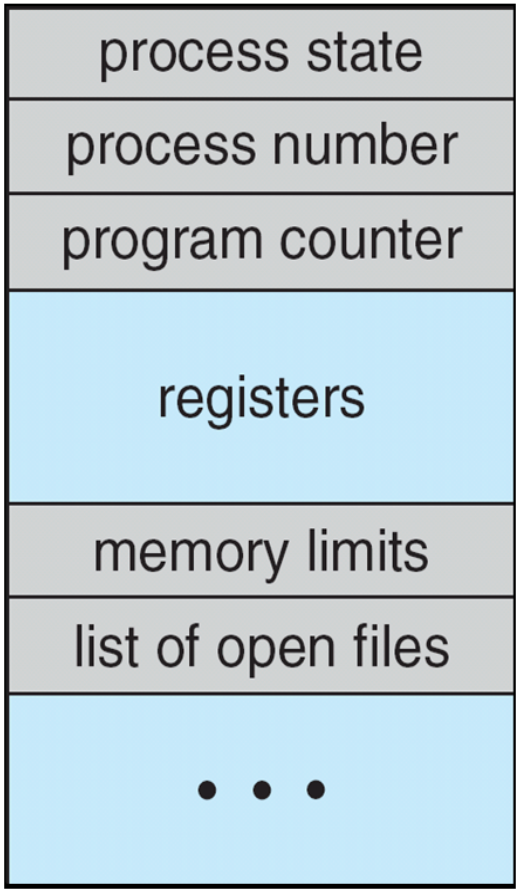


Process Attributes

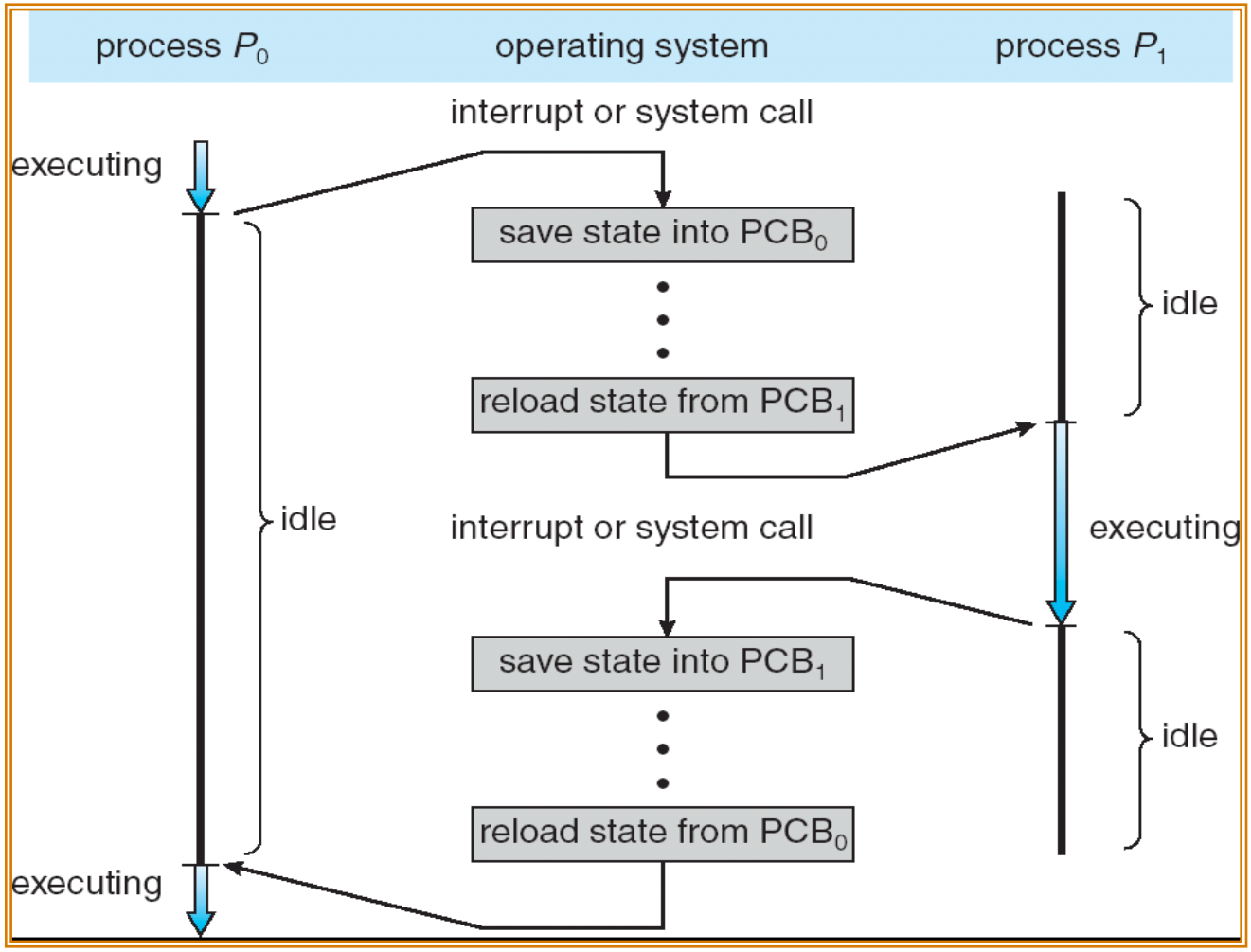
- Process identification
  - Process identifier
  - Parent process identifier
  - User identifier
- Processor state information (contents of processor registers).
  - User-visible registers
  - Control and status registers: **program counter**, etc
  - Stack pointers
- Process control information
  - Process state
  - Memory pointers
  - Interprocess communication, etc.

Process Control Block (PCB)

- PCB (representation of process by OS) contains
  - Process state
  - Process identifier (number)
  - Program counter
  - CPU registers (context)
  - CPU scheduling information
  - Memory-management information
  - Accounting information
  - I/O status information

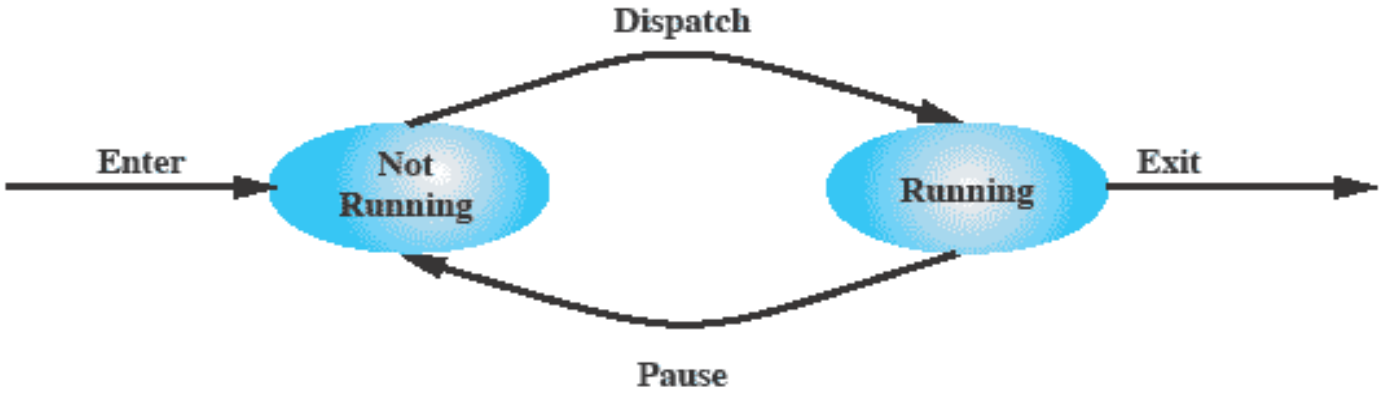


Context Switch



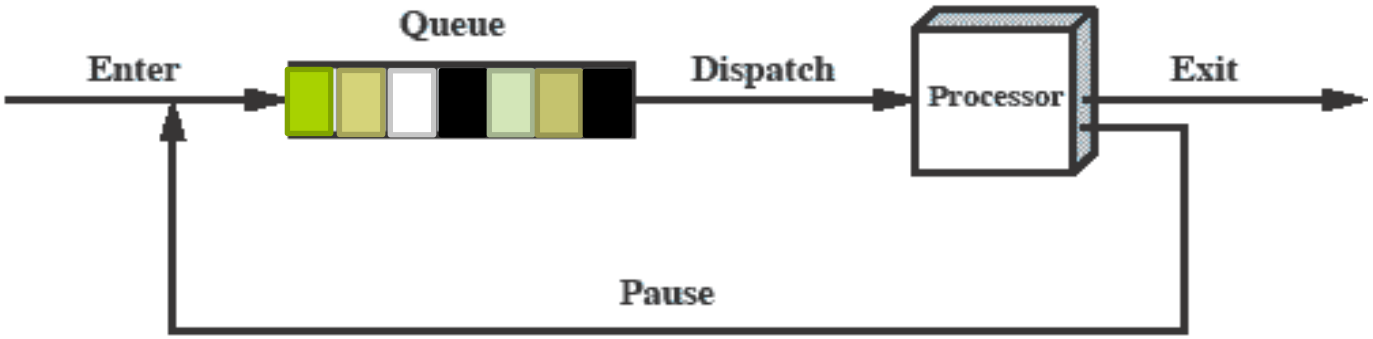
Two-State Process Model

- 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

Five-state Process Model