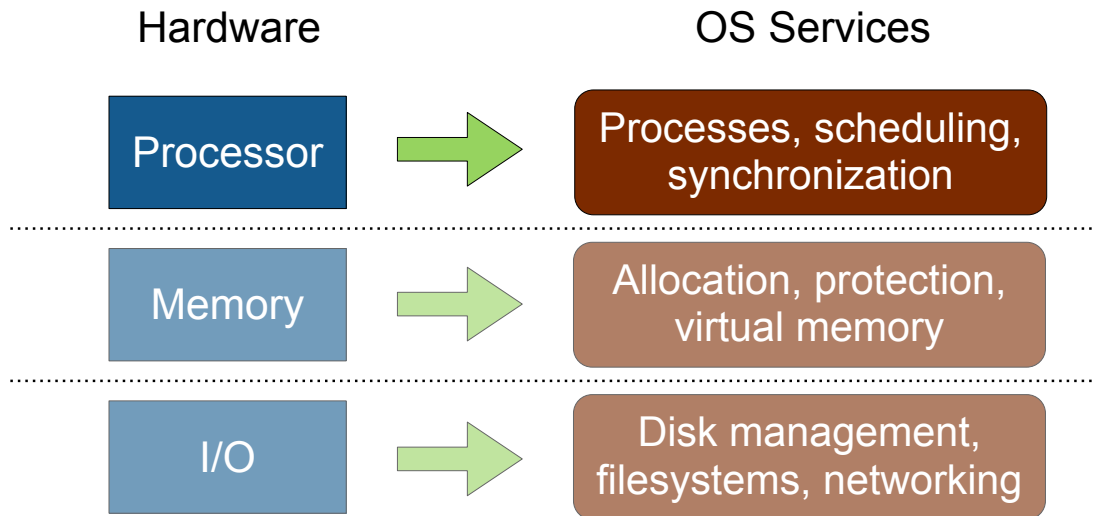
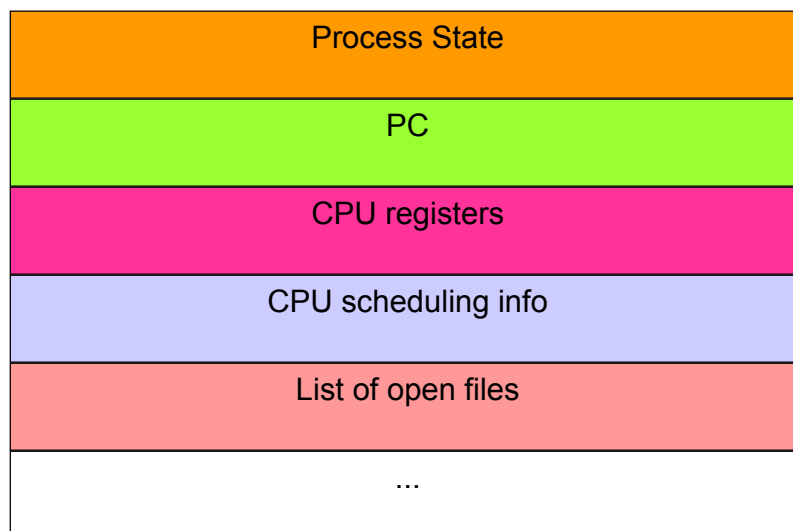


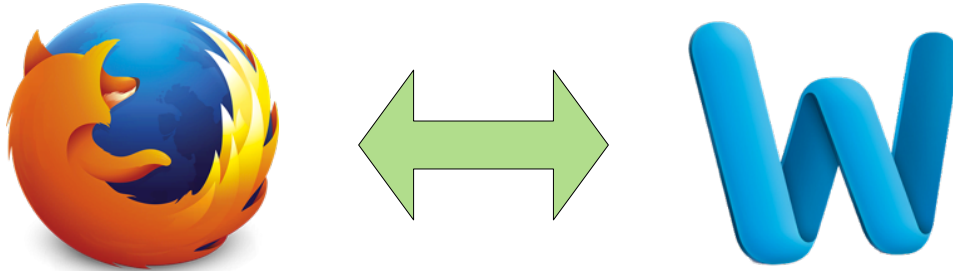
Processes



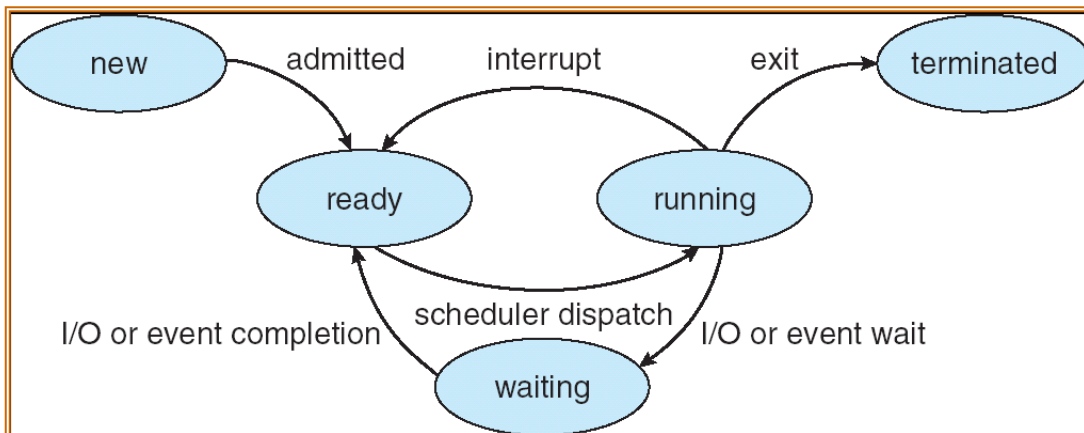
Process Control Block (PCB)



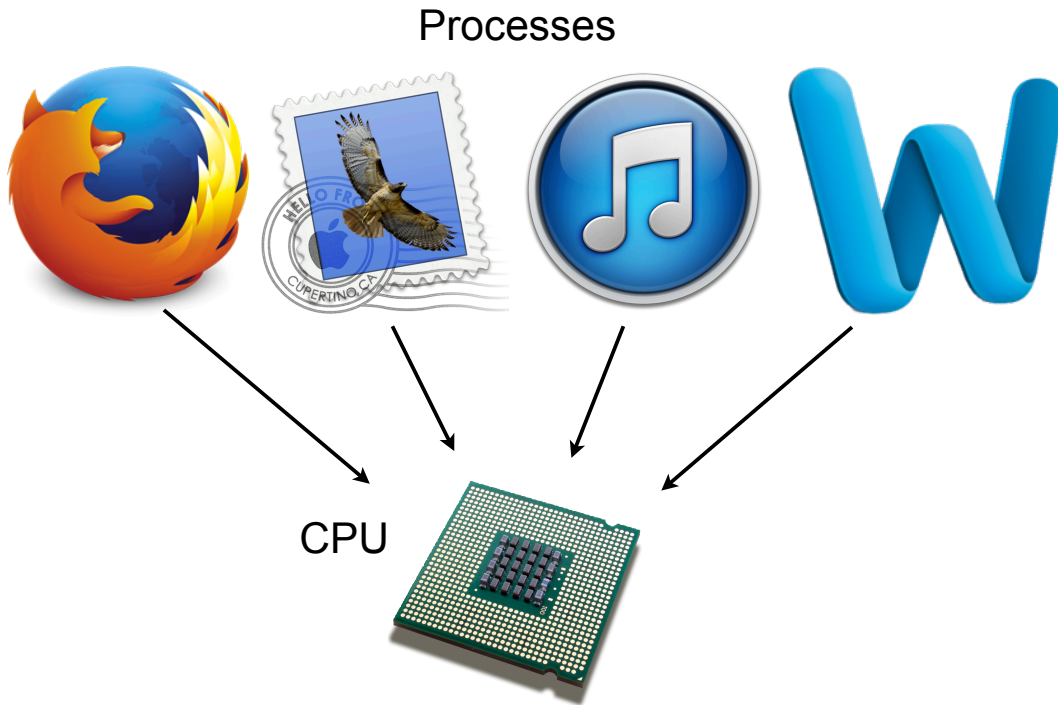
Context Switching



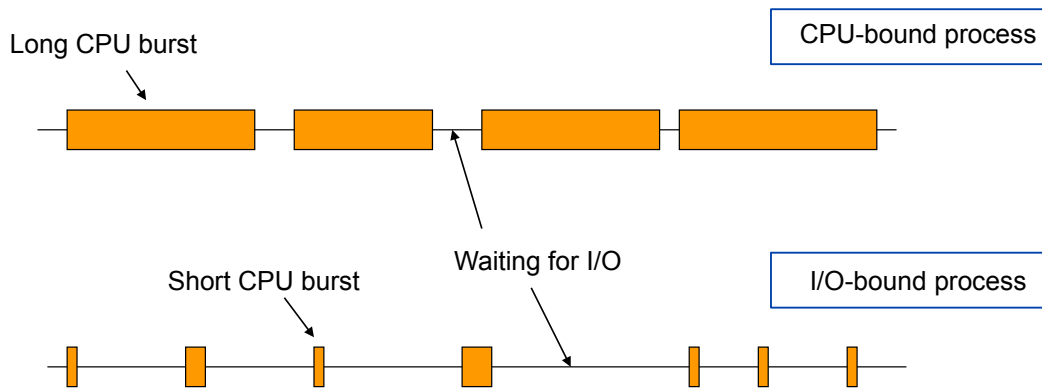
Process Execution States



CPU Scheduling



CPU and I/O Bursts



Multilevel Feedback Queues

	Priority	Time Slice				
<table border="1"><tr><td></td><td>G</td><td>F</td><td>A</td></tr></table>		G	F	A	1	1
	G	F	A			
<table border="1"><tr><td></td><td></td><td>E</td></tr></table>			E	2	2	
		E				
<table border="1"><tr><td></td><td>D</td><td>B</td></tr></table>		D	B	3	4	
	D	B				
<table border="1"><tr><td></td><td>C</td></tr></table>		C	4	8		
	C					

CPU Scheduling Summary

First Come First Serve (FCFS)

Shortest Job First (SJF)

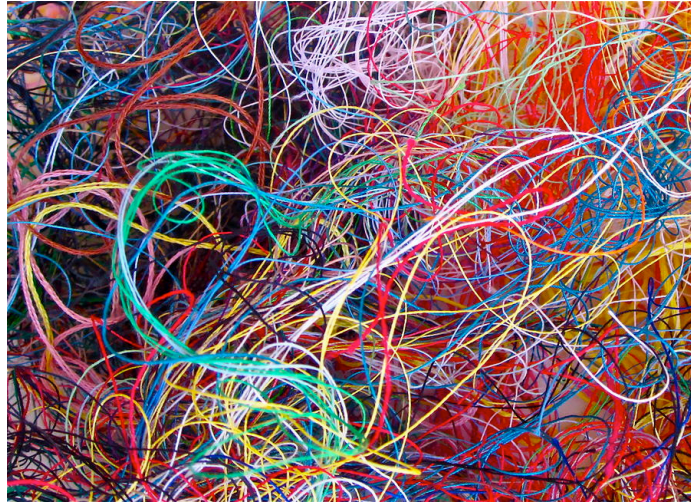
Shortest Remaining Time First (SRTF)

Round-Robin (RR)

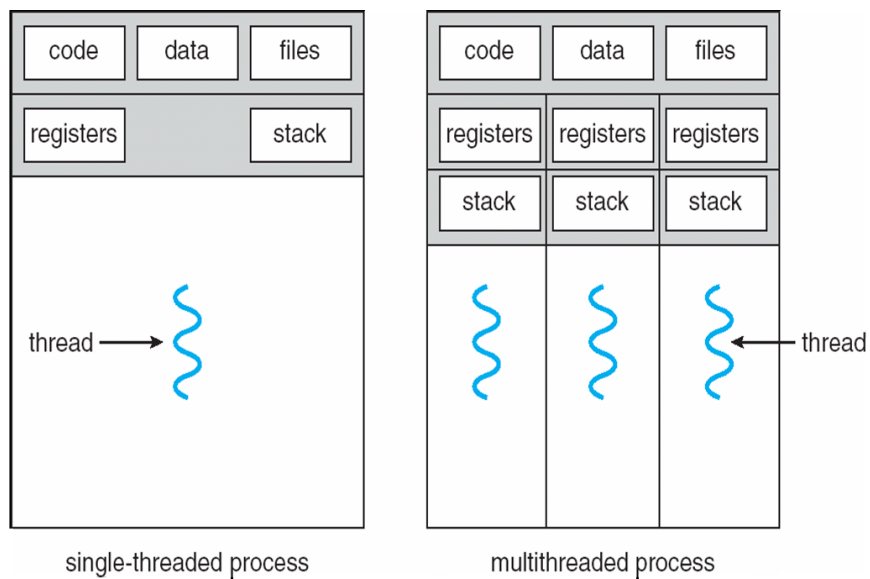
Multilevel Feedback Queues (MLFQ)

... more advanced schedulers ...

Threads



Multithreaded Processes



Thread APIs

POSIX Threads (pthreads):

```
pthread_create(&tid, NULL, my_fun, &param); // thread runs my_fun
```

Windows Threads:

```
ThreadHandle = CreateThread(NULL, 0, MyFun, &Param, 0, &ThreadID);
```

Java Threads:

```
Thread t = new Thread(new MyRunnable(param));  
t.start(); // start the thread running MyRunnable.run()
```

User-Level Threads

