



Operating Systems

Processes-Part1

Seyyed Ahmad Javadi

sajavadi@aut.ac.ir

Fall 2023

Copyright Notice

Slides are based on the slides of the main **textbook**.

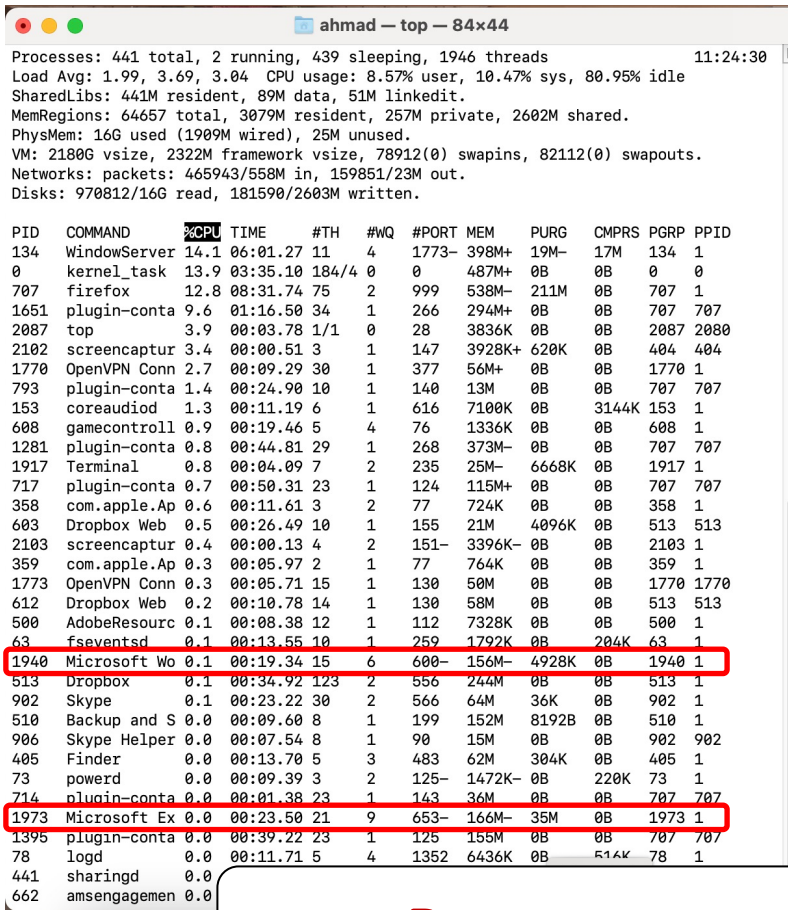
Silberschatz

<https://www.os-book.com/OS10/slide-dir/index.html>



Process Concept

- An OS executes a variety of programs that run as a **process**.



Processes: 441 total, 2 running, 439 sleeping, 1946 threads 11:24:30
Load Avg: 1.99, 3.69, 3.04 CPU usage: 8.57% user, 10.47% sys, 80.95% idle
SharedLibs: 441M resident, 89M data, 51M linkedit.
MemRegions: 64657 total, 3079M resident, 257M private, 2602M shared.
PhysMem: 16G used (1909M wired), 25M unused.
VM: 2180G vsize, 2322M framework vsize, 78912(0) swapins, 82112(0) swapouts.
Networks: packets: 465943/558M in, 159851/23M out.
Disks: 970812/16G read, 181590/2603M written.

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPRS	PGRP	PPID
134	WindowServer	14.1	06:01.27	11	4	1773-	398M+	19M-	17M	134	1
0	kernel_task	13.9	03:35.10	184/4	0	0	487M+	0B	0B	0	0
707	firefox	12.8	08:31.74	75	2	999	538M-	211M	0B	707	1
1651	plugin-conta	9.6	01:16.50	34	1	266	294M+	0B	0B	707	707
2087	top	3.9	00:03.78	1/1	0	28	3836K	0B	0B	2087	2080
2102	screencaptur	3.4	00:00.51	3	1	147	3928K+	620K	0B	404	404
1770	OpenVPN Conn	2.7	00:09.29	30	1	377	56M+	0B	0B	1770	1
793	plugin-conta	1.4	00:24.90	10	1	140	13M	0B	0B	707	707
153	coreaudioid	1.3	00:11.19	6	1	616	7100K	0B	3144K	153	1
608	gamecontroll	0.9	00:19.46	5	4	76	1336K	0B	0B	608	1
1281	plugin-conta	0.8	00:44.81	29	1	268	373M-	0B	0B	707	707
1917	Terminal	0.8	00:04.09	7	2	235	25M-	6668K	0B	1917	1
717	plugin-conta	0.7	00:50.31	23	1	124	115M+	0B	0B	707	707
358	com.apple.Ap	0.6	00:11.61	3	2	77	724K	0B	0B	358	1
603	Dropbox Web	0.5	00:26.49	10	1	155	21M	4096K	0B	513	513
2103	screencaptur	0.4	00:00.13	4	2	151-	3396K-	0B	0B	2103	1
359	com.apple.Ap	0.3	00:05.97	2	1	77	764K	0B	0B	359	1
1773	OpenVPN Conn	0.3	00:05.71	15	1	130	50M	0B	0B	1770	1770
612	Dropbox Web	0.2	00:10.78	14	1	130	58M	0B	0B	513	513
500	AdobeResourc	0.1	00:08.38	12	1	112	7328K	0B	0B	500	1
63	fseventsfd	0.1	00:13.55	10	1	259	1792K	0B	204K	63	1
1940	Microsoft Wo	0.1	00:19.34	15	6	600-	156M-	4928K	0B	1940	1
513	Dropbox	0.1	00:34.92	123	2	556	244M	0B	0B	513	1
902	Skype	0.1	00:23.22	30	2	566	64M	36K	0B	902	1
510	Backup and S	0.0	00:09.60	8	1	199	152M	8192B	0B	510	1
906	Skype Helper	0.0	00:07.54	8	1	90	15M	0B	0B	902	902
405	Finder	0.0	00:13.70	5	3	483	62M	304K	0B	405	1
73	powerd	0.0	00:09.39	3	2	125-	1472K-	0B	220K	73	1
714	plugin-conta	0.0	00:01.38	23	1	143	36M	0B	0B	707	707
1973	Microsoft Ex	0.0	00:23.50	21	9	653-	166M-	35M	0B	1973	1
1395	plugin-conta	0.0	00:39.22	23	1	125	155M	0B	0B	707	707
78	logd	0.0	00:11.71	5	4	1352	6436K	0B	516K	78	1
441	sharingd	0.0									
662	amsengagemen	0.0									

Document2

Page 1 of 1 0 words

Book2

Home Insert Draw Page Layout Formulas Data Review

Font Alignment Number

Conditional Formatting

Format as Table

Cell Styles

Cells Editing

A1

fx

A B C D E F G H I J

1

2

3

4

5

6

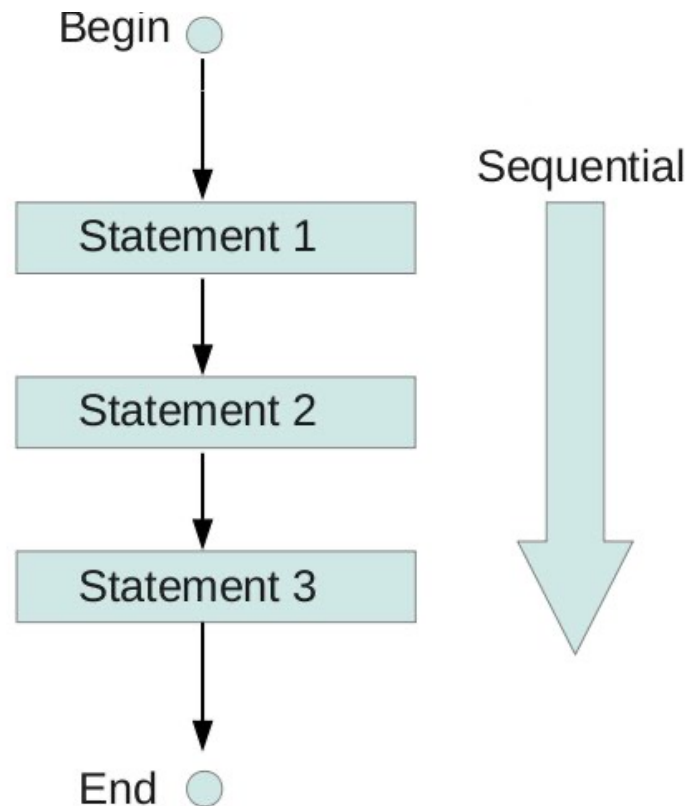
Sheet1

100%

Process, a program in execution

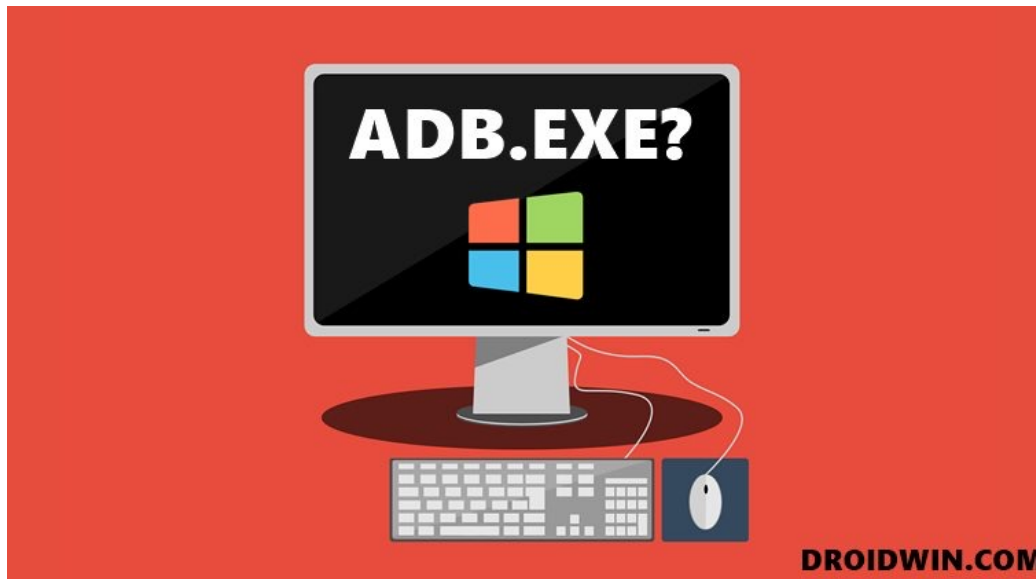
Process Concept (cont.)

- Process execution must progress in ***sequential fashion***.
 - ***No parallel execution*** of instructions of a single process.



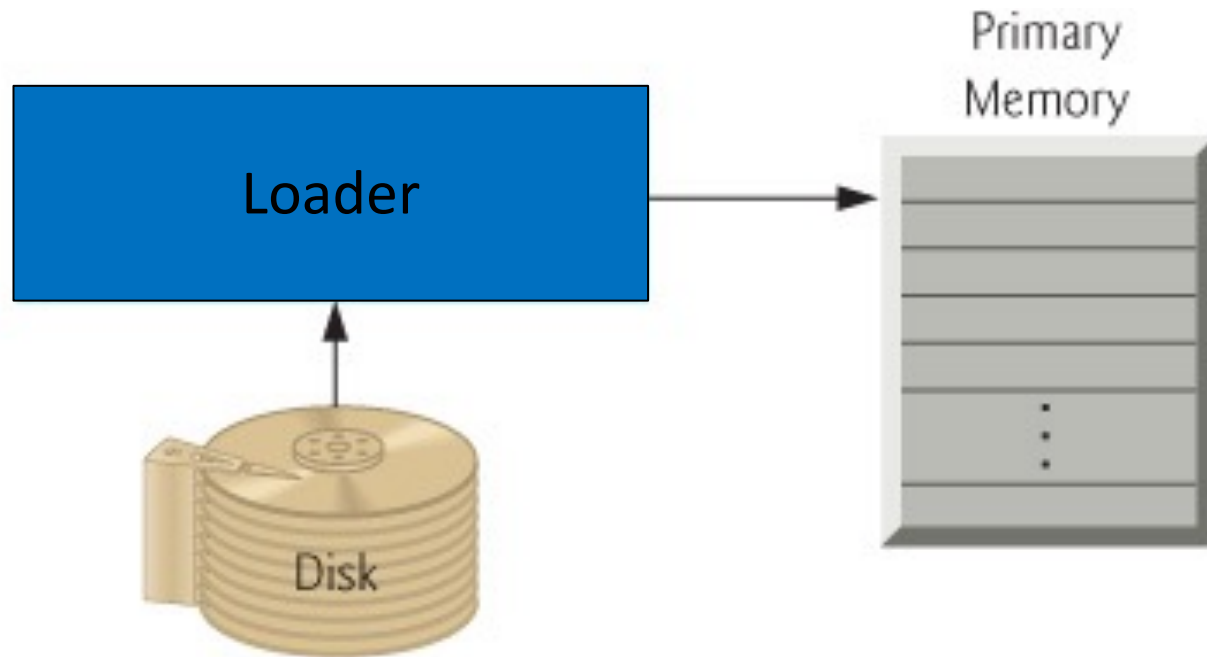
Process versus Program

- Program is **passive** entity stored on disk (*executable file*).



Process versus Program (cont.)

- **Process is active.**
 - **Program becomes process** when an executable file is loaded into memory.



Process versus Program (cont.)

- Execution of program started via:
 - GUI mouse clicks
 - Command line entry of its name
 - Etc.

- One program can be several processes
 - How?
 - Consider multiple users executing the same program.

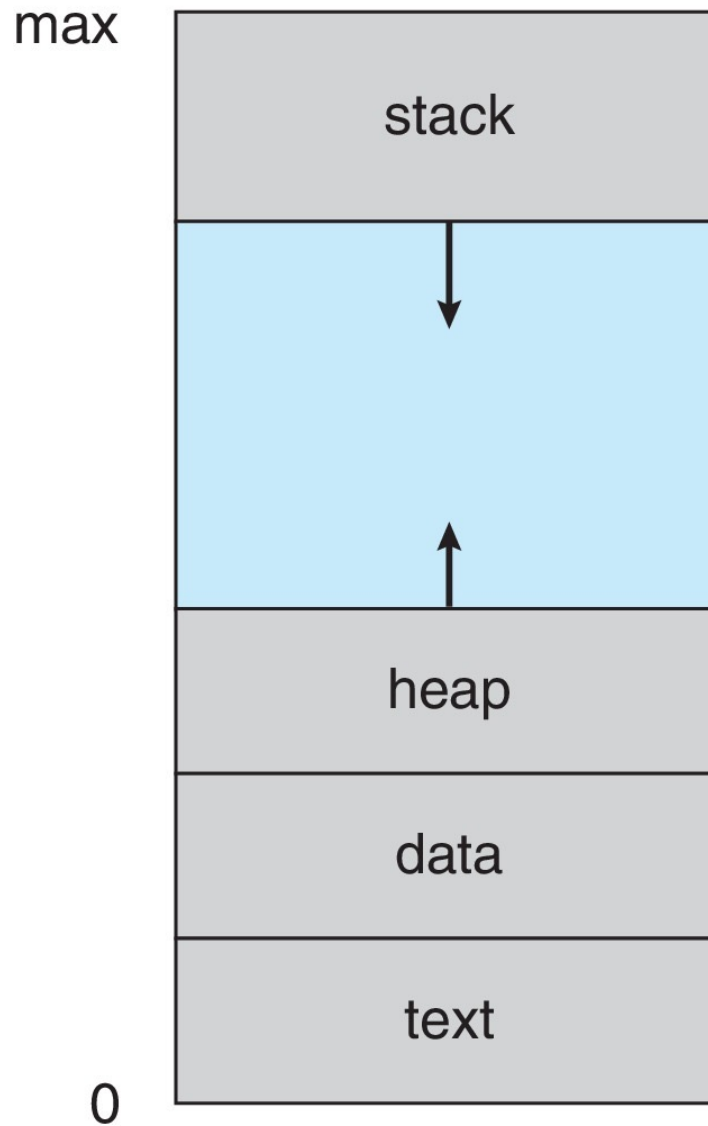


Multiple Parts of Process

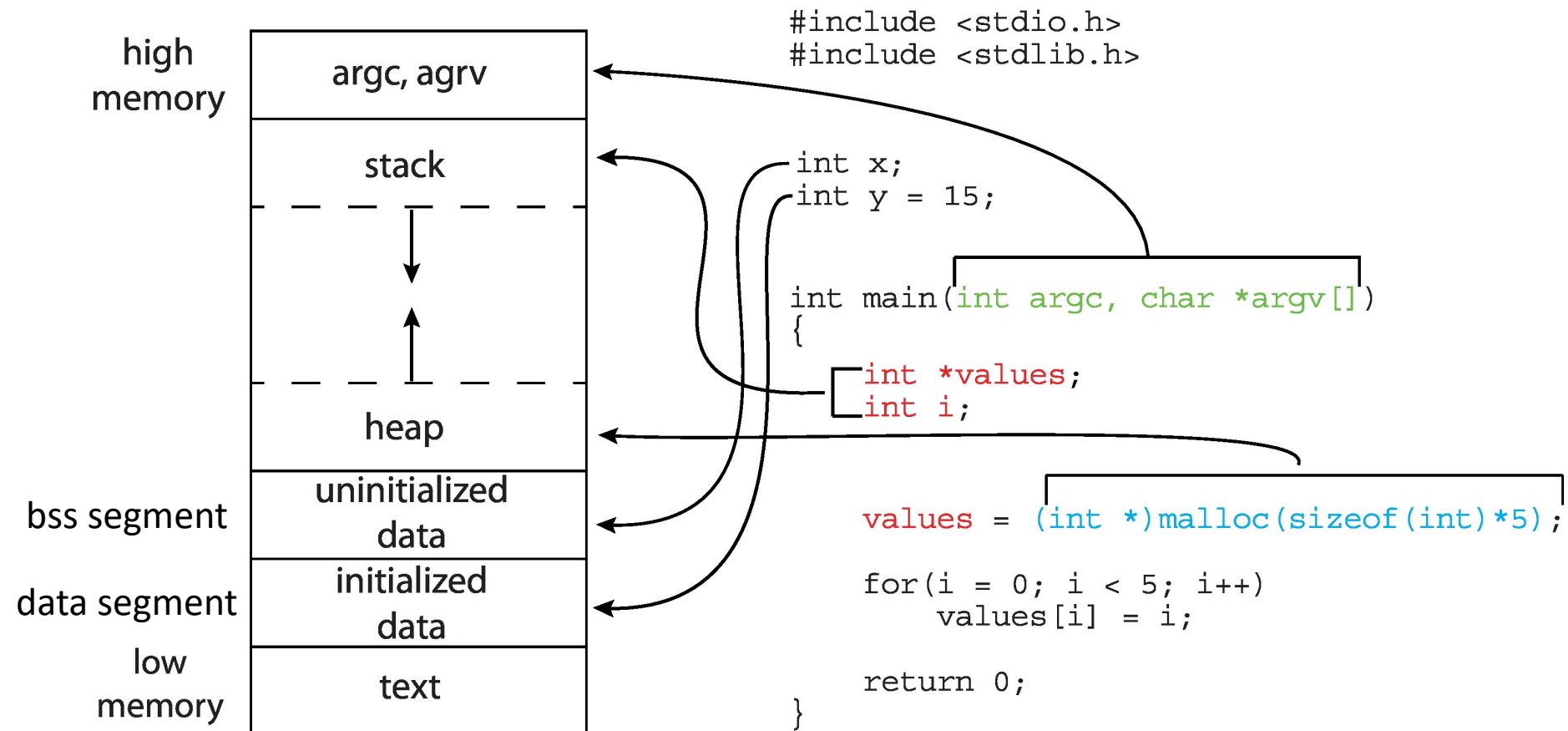
- The program code, also called *text section*
- Current activity including *program counter*, processor registers
- *Stack* containing temporary data
 - Function parameters, return addresses, local variables
- *Data section* containing global variables
- *Heap* containing memory *dynamically* allocated during run time



Process in Memory



Memory Layout of a C Program



Some Links

<https://www.geeksforgeeks.org/memory-layout-of-c-program/>

<https://stackoverflow.com/questions/10315759/data-section-size-in-size-command-on-mac>



Process State

- As a process executes, it changes **state**



- **New:** The process is being created
- **Running:** Instructions are being executed
- **Waiting:** The process is waiting for some event to occur
- **Ready:** The process is waiting to be assigned to a processor
- **Terminated:** The process has finished execution

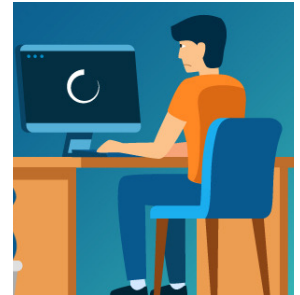
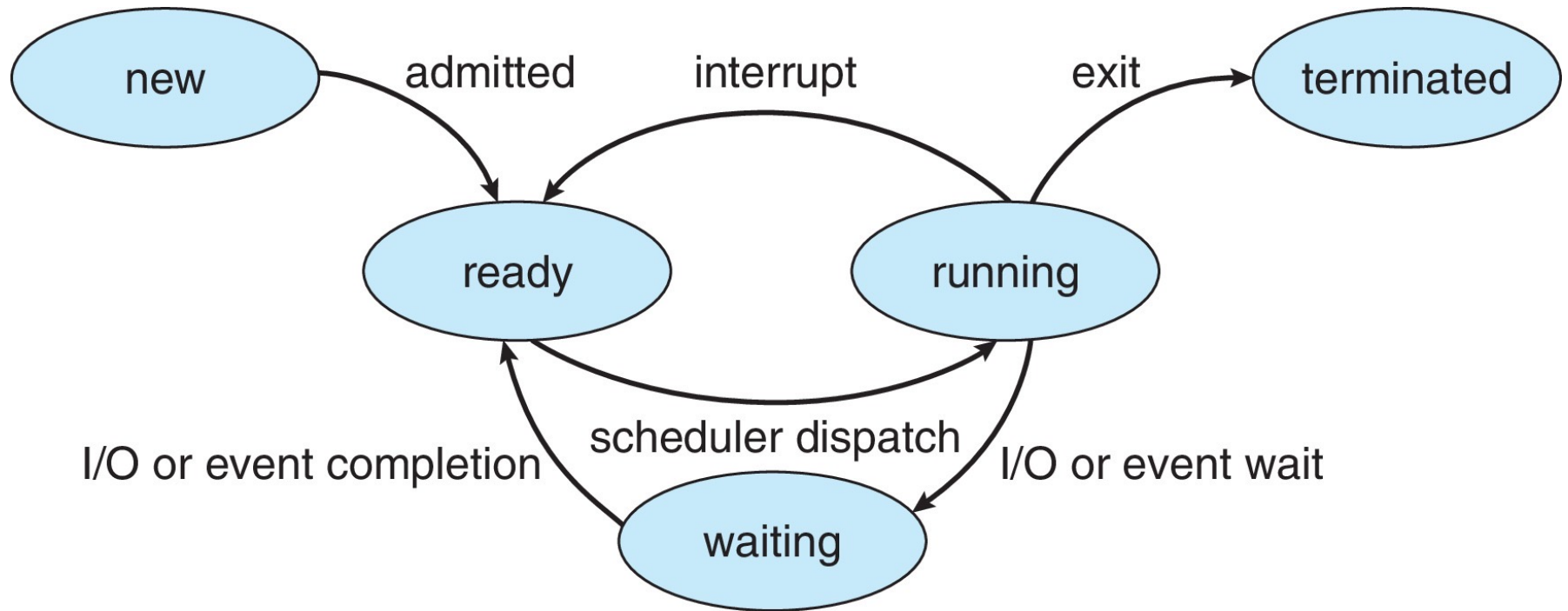


Diagram of Process State



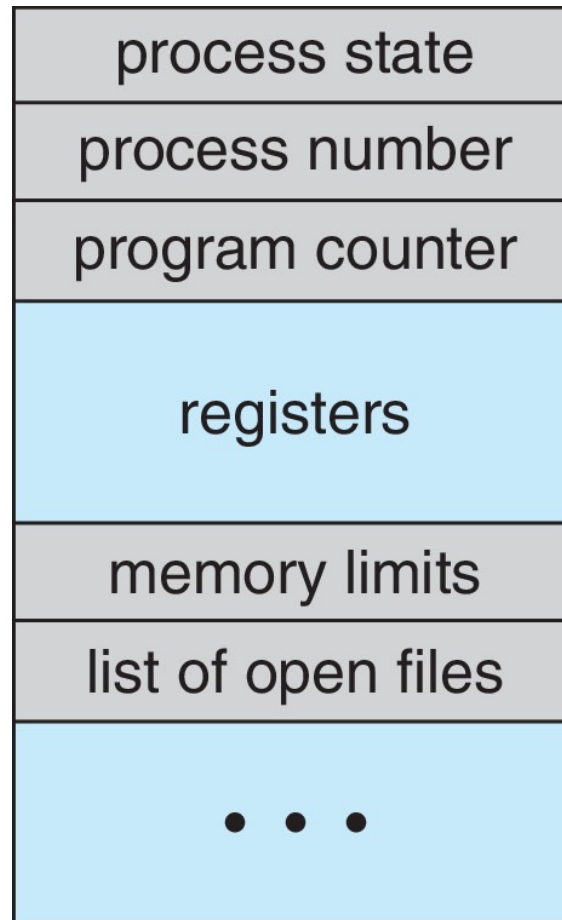
Process Control Block (PCB)

Information associated with each process

- **Process state**: running, waiting, etc.
- **Program counter**: location of instruction to next execute.
- **CPU registers**: contents of all process-centric registers.
- **CPU scheduling information**: priorities, scheduling queue pointers.
- **Memory-management information**: allocated memory
- **Accounting information**: CPU used, clock time elapsed since start, etc
- **I/O status information**: allocated I/O devices, list of open files.

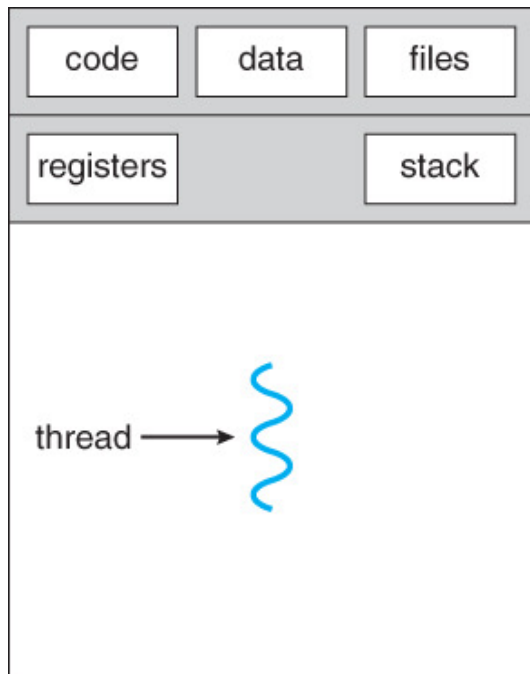


Process Control Block (PCB) (cont.)

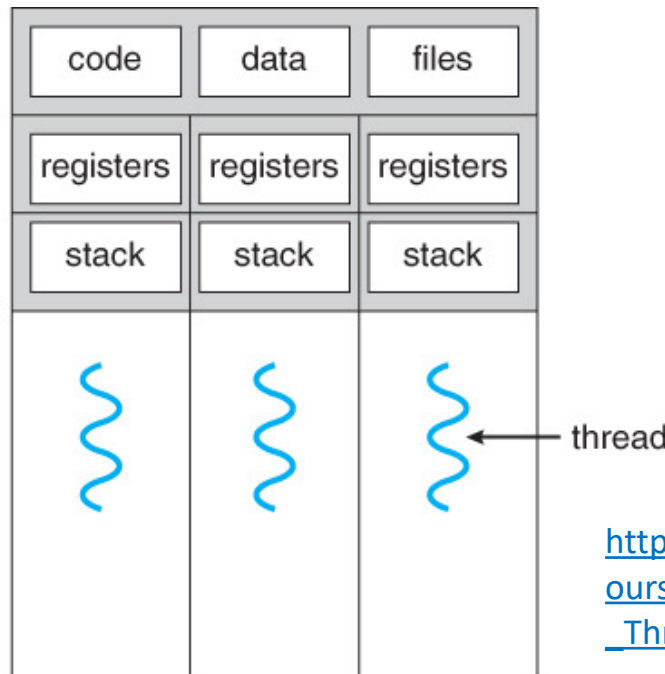


Threads

- So far, process has ***a single thread of execution***.
- Consider having ***multiple program counters*** per process.
 - Multiple locations can execute at once
 - ▶ Multiple threads of control -> **thread**



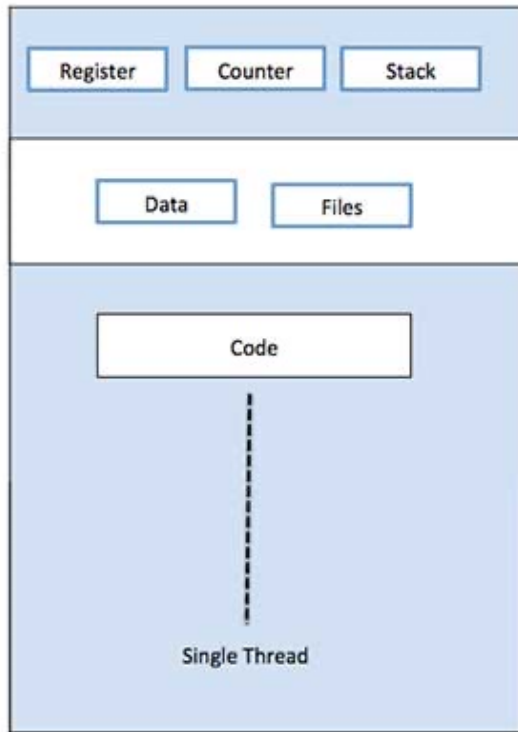
single-threaded process



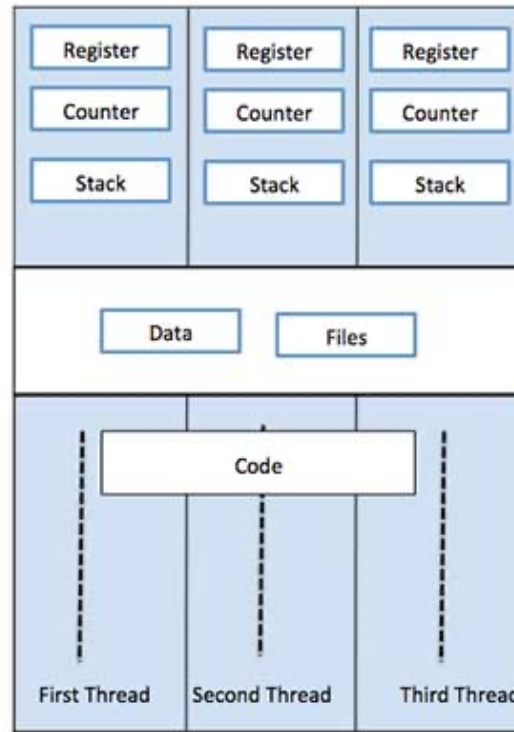
multithreaded process

https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/4_Threads.html

Threads



Single Process P with single thread



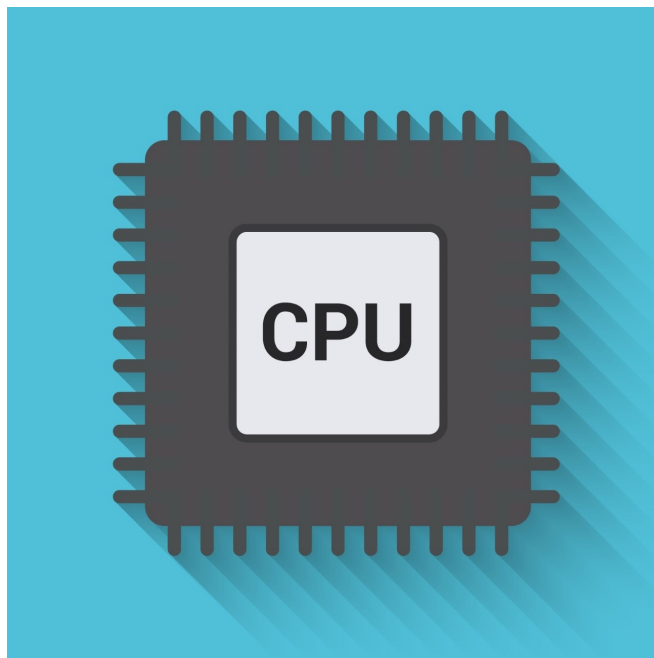
Single Process P with three threads

https://www.tutorialspoint.com/operating_system/os_multi_threading.htm

- Must then have ***storage for thread details***
 - Multiple program counters in PCB.
- Explore in detail in ***Chapter 4***.

Process Scheduling

- ***Process scheduler*** selects among available processes for next execution on CPU core.
- **Goal: Maximize CPU use, quickly switch processes onto CPU core.**

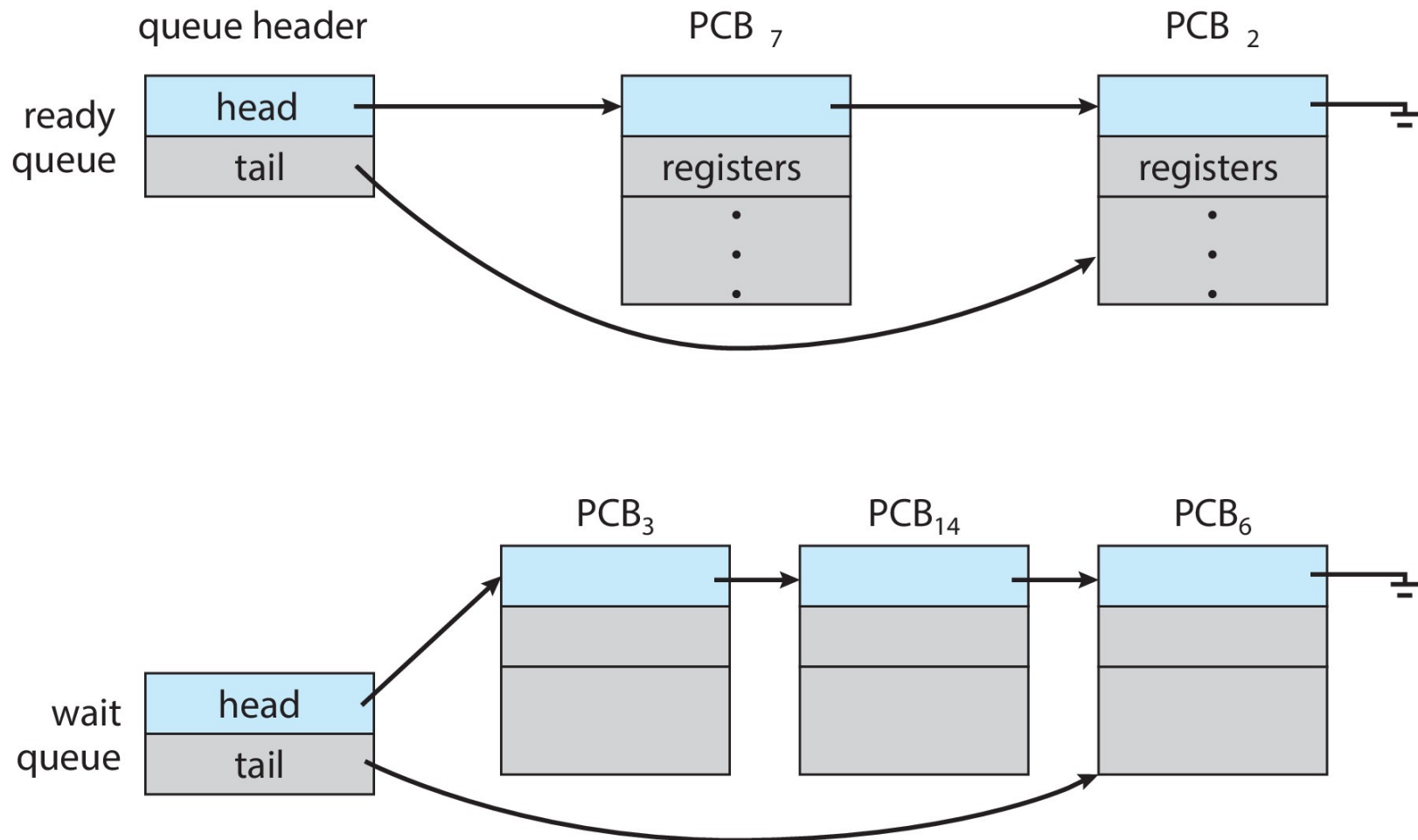


Process Scheduling (cont.)

- Maintains *scheduling queues* of processes
 - **Ready queue**
 - ▶ Set of all processes residing in main memory, ready and waiting to execute.
 - **Wait queues**
 - ▶ Set of processes waiting for an event (i.e., I/O)
- Processes *migrate* among the various queues.



Ready and Wait Queues



Representation of Process Scheduling

