University of Glasgow

Systems Programming – Part 2
Concurrent Systems Programming

Dr Lauritz Thamsen
lauritz.thamsen@glasgow.ac.uk
https://lauritzthamsen.org

University of Glasgow

Topics of Part 2 of SP(GA)

- Intro to Concurrency (with Processes and Threads)
- Process/Thread Synchronisation
- **More on Process Management (from an OS Perspective)**
- Concurrency Beyond Threads & Limits of Scalability
- Virtual Memory & Levels of Storage

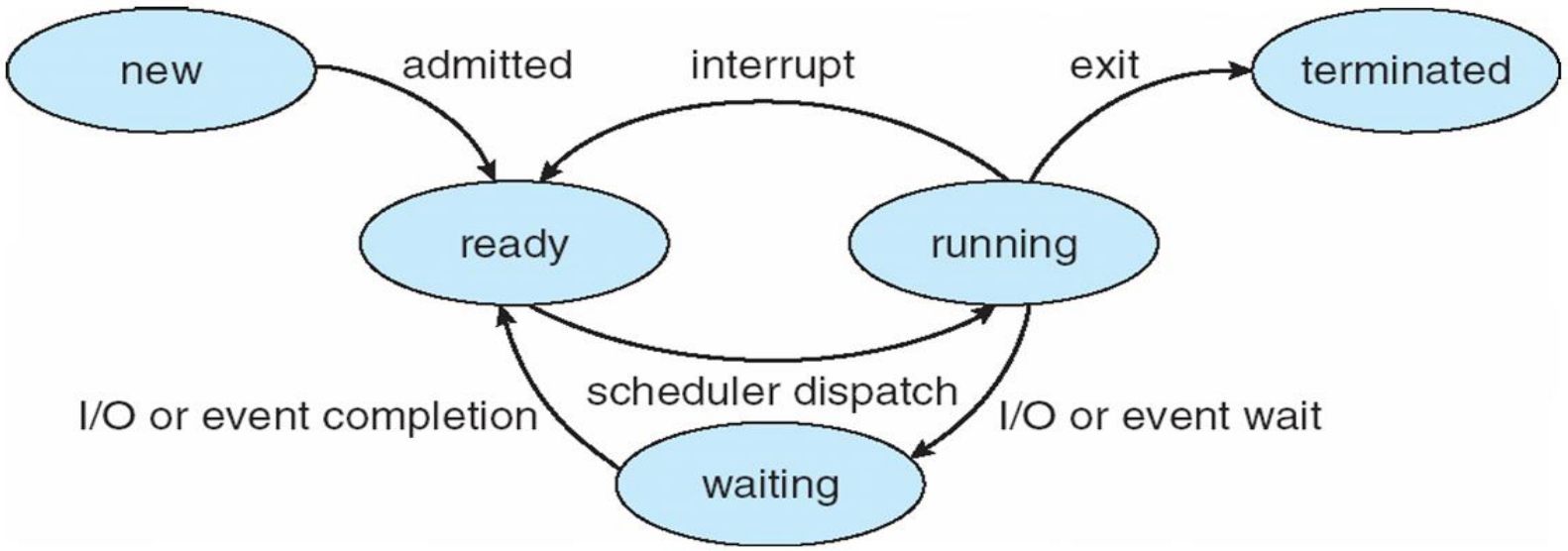
University of Glasgow

Lecture Outline

- The Life of a Process
- Recap on Scheduling
- To Try at Home (and OS(H))

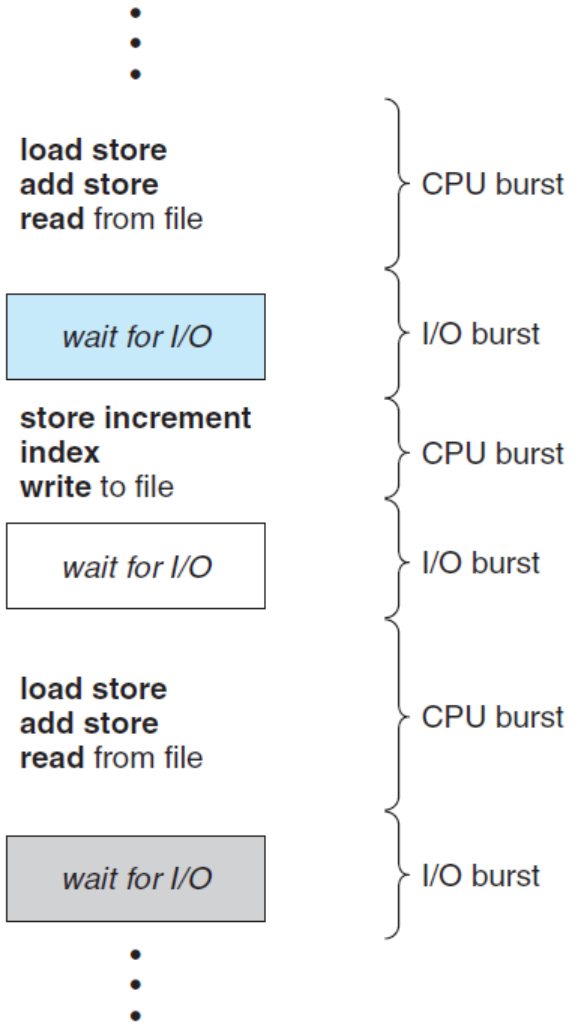
Recap: Life of a Process

- Process states and transitions:



Typical Life of a Process

CPU bound?

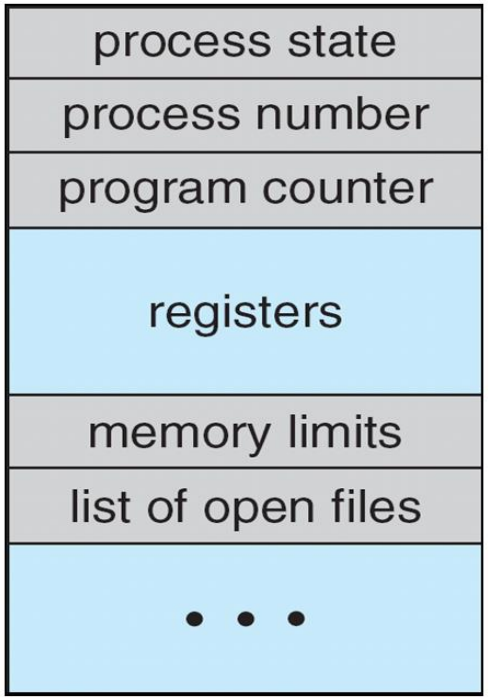


Source: A. Silberschatz, "Operating System Concepts", 9th Ed., 2012.

IO bound?

Recap: Process Control Block (PCB)

- Information associated with each process
  - Also known as Process Table Entry



University of Glasgow

Process Creation

- Parent process creates child processes
  - A “process tree” captures parent-child relations
  - fork(2), exec(3), wait(2)

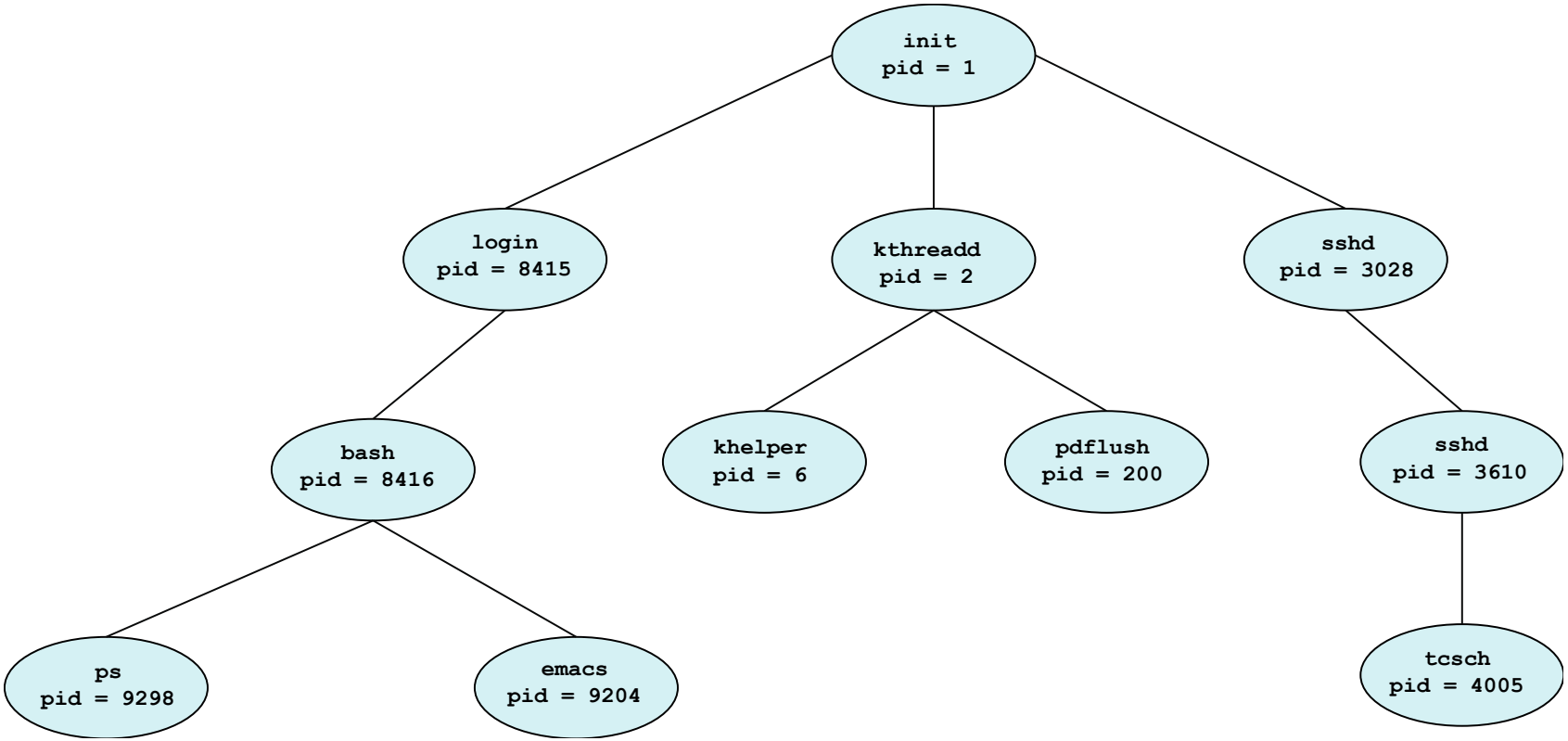
- Execution options
  - Parent and children execute concurrently
  - Parent waits for children to terminate

University of Glasgow

Fork, Exec, and Wait on a Unix-alike System in C

```c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
 pid t pid;

 /* fork a child process */
 pid = fork();

 if (pid < 0) { /* error occurred */
  fprintf(stderr, "Fork Failed"); return 1;
 } else if (pid == 0) { /* child process */
  execlp("/bin/ls", "ls", NULL);
 } else { /* parent process */
  /* parent will wait for the child to complete */
  wait(NULL);
  printf("Child Complete");
 }

 return 0;
}
```

Example: Linux Process Tree



University of Glasgow

Lecture Outline

- The Life of a Process
- **Recap on Scheduling**
- To Try at Home (and OS(H))

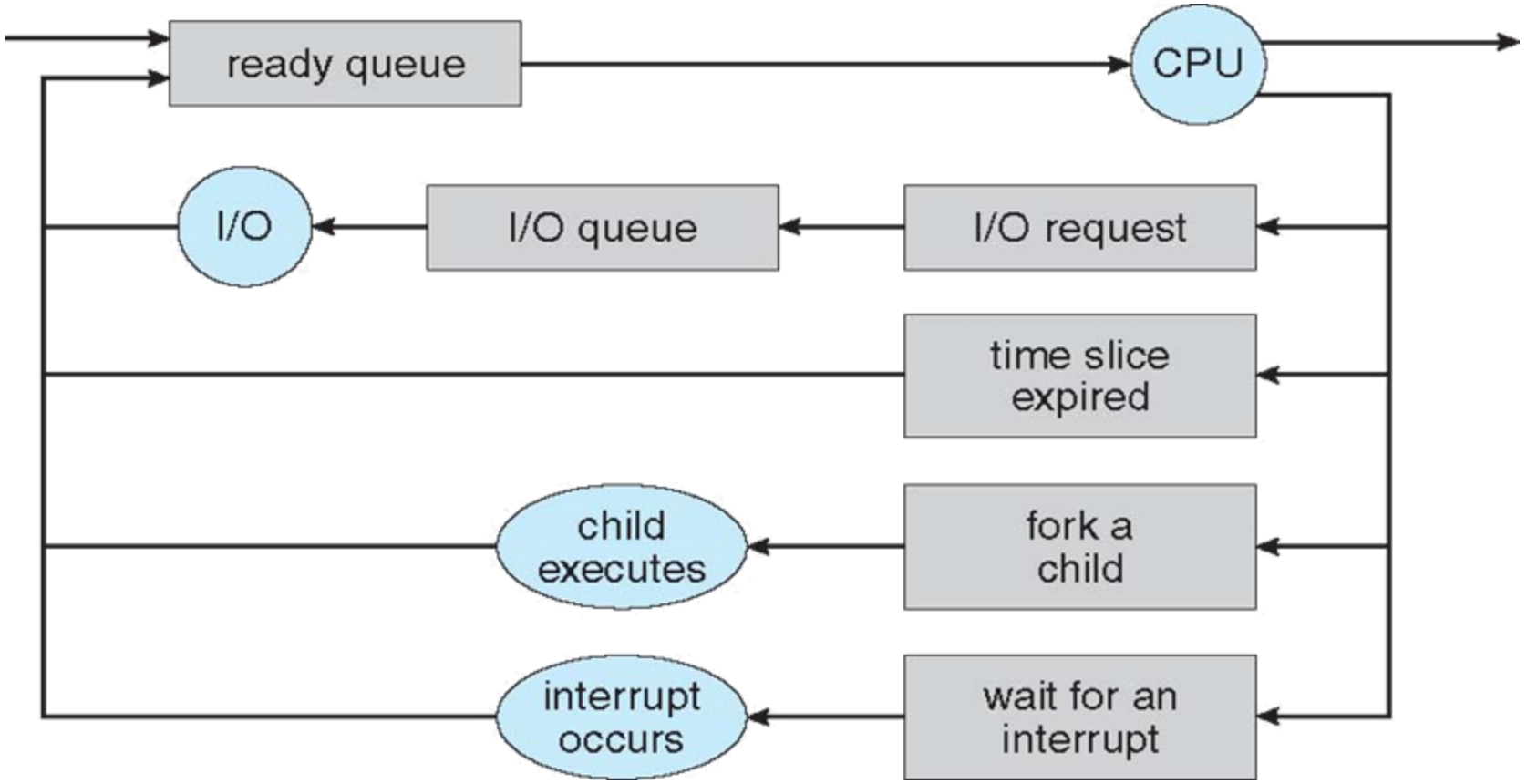
University of Glasgow

Recap: Process Scheduling

- (CPU-)Scheduling's main goal: Maximize CPU utilization

- Several queues of processes need to be maintained:
  - **Job queue** (all processes in the system)
  - **Ready queue** (processes in memory and ready to execute)
  - **Device queues** (processes waiting on some device; one such queue per device)
  - Processes can migrate among the various queues

Queue-Based Process Scheduling



Grey boxes: actions/queues
Blue circles: queue servers

University of Glasgow

Recap: CPU Scheduler

- CPU scheduling decisions may take place when a process:
  1. Switches from running to waiting state (e.g., IO request, sleep, etc.)
  2. Switches from running to ready state (e.g., interrupt occurs)
  3. Switches from waiting to ready (e.g., completion of IO)
  4. Terminates

- **Non-preemptive scheduling**: Let the process give up (yield) the CPU (Cases 1 and 4)
- **Preemptive scheduling**: The Scheduler decides when a process yields the CPU (all cases)

University of Glasgow

Dispatcher

- Gives control of the CPU to the process selected
- Needs to:
  - Switch context (load registers, PCB, etc.)
  - Switch to user mode
  - Jump to the proper instruction in the user program to continue execution
- Latency?
  - Note: context switch is pure overhead!

University of Glasgow

Recap: CPU Schedulers

- First-Come, First-Served (FCFS or FIFO)
- Priority based (preemptive and non-preemptive)
- Shortest Job First (SJF)
- Shortest Remaining Time First (SRTF)
- Round-Robin (RR)

Covered in CANS and OS(H)! ☺

University of Glasgow

Lecture Outline

- The Life of a Process
- Recap on Scheduling
- **To Try at Home (and OS(H))**

University of Glasgow

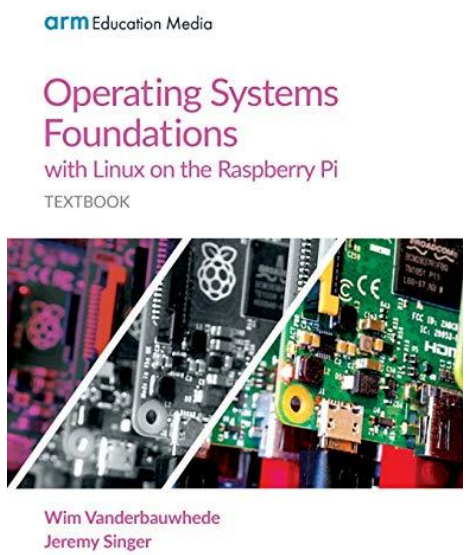
To Try at Home!

- Investigate which tools allow you to see running processes on your OS
  - Which process is using the most memory and which the most CPU? Disk I/O? Network?
  - How many threads does a specific process have? Which user started a process? Which process has been running for the longest time?
- How can you find a specific process?
- Can you stop/kill a specific process while it runs?
  - Note: Please only try this, if you know it's fine to stop a particular process

More Also in OS(H)

→ COMPSCI4011 Operating Systems (H)

- **School:** School of Computing Science
- **Credits:** 10
- **Level:** Level 4 (SCQF level 10)
- **Typically Offered:** Semester 2



- https://www.gla.ac.uk/coursecatalogue/course/?code=COMPSCI4011

University of Glasgow

Recommended Reading

- Silberschatz, Galvin, Gagne, Operating Systems Concepts, Sections 3.1-3, 4.1, and 6.1-3