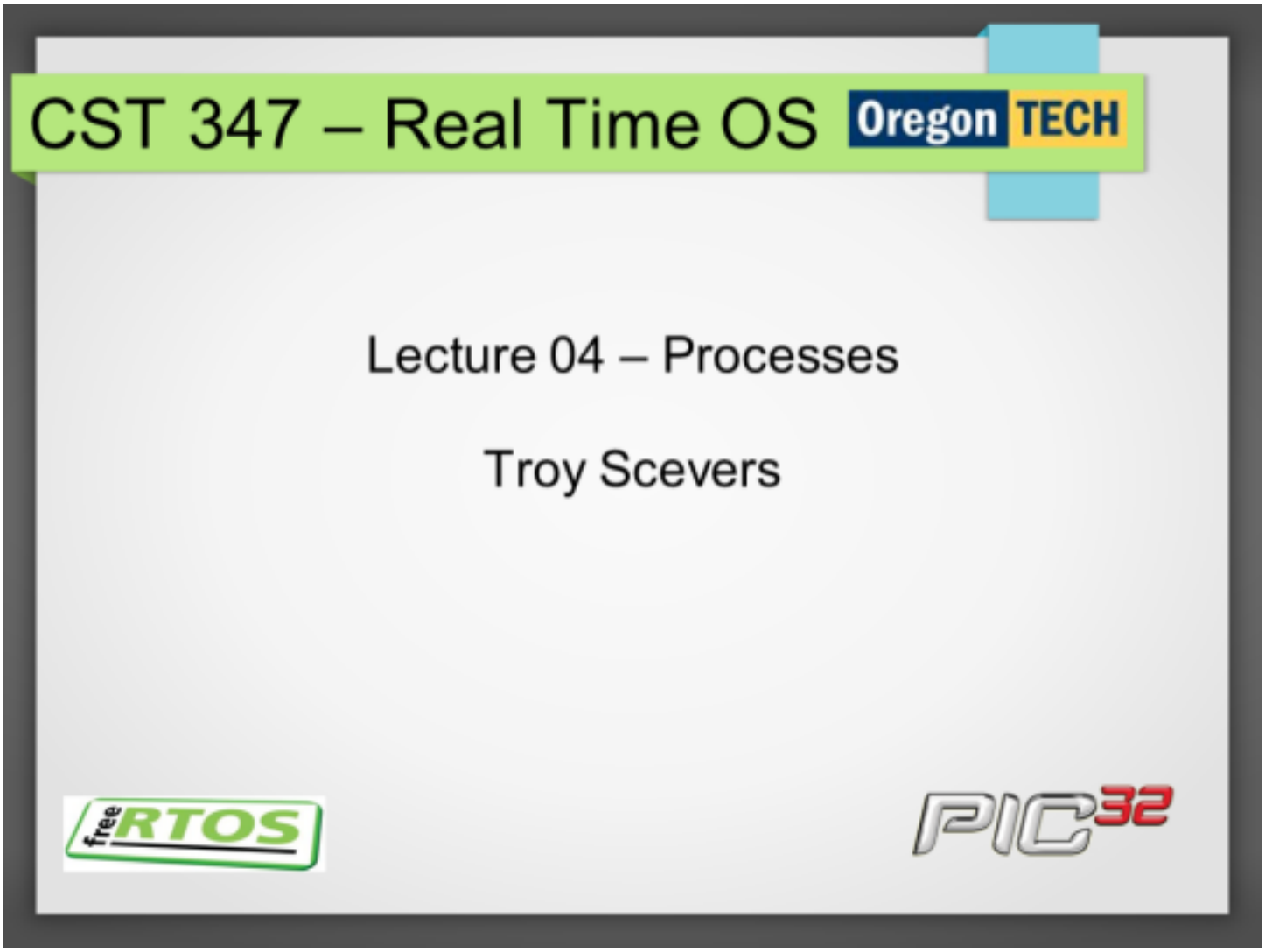
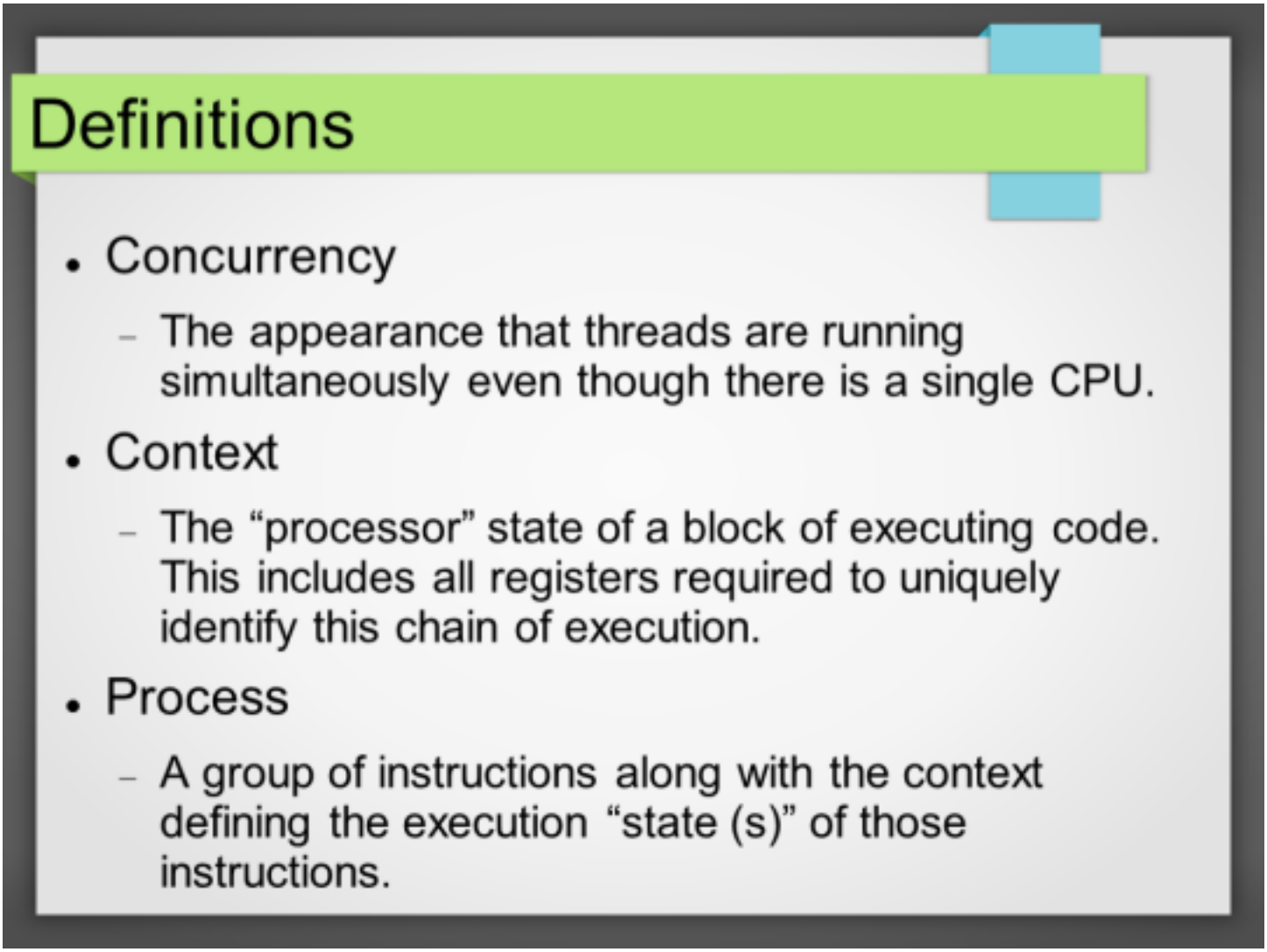
Public Page

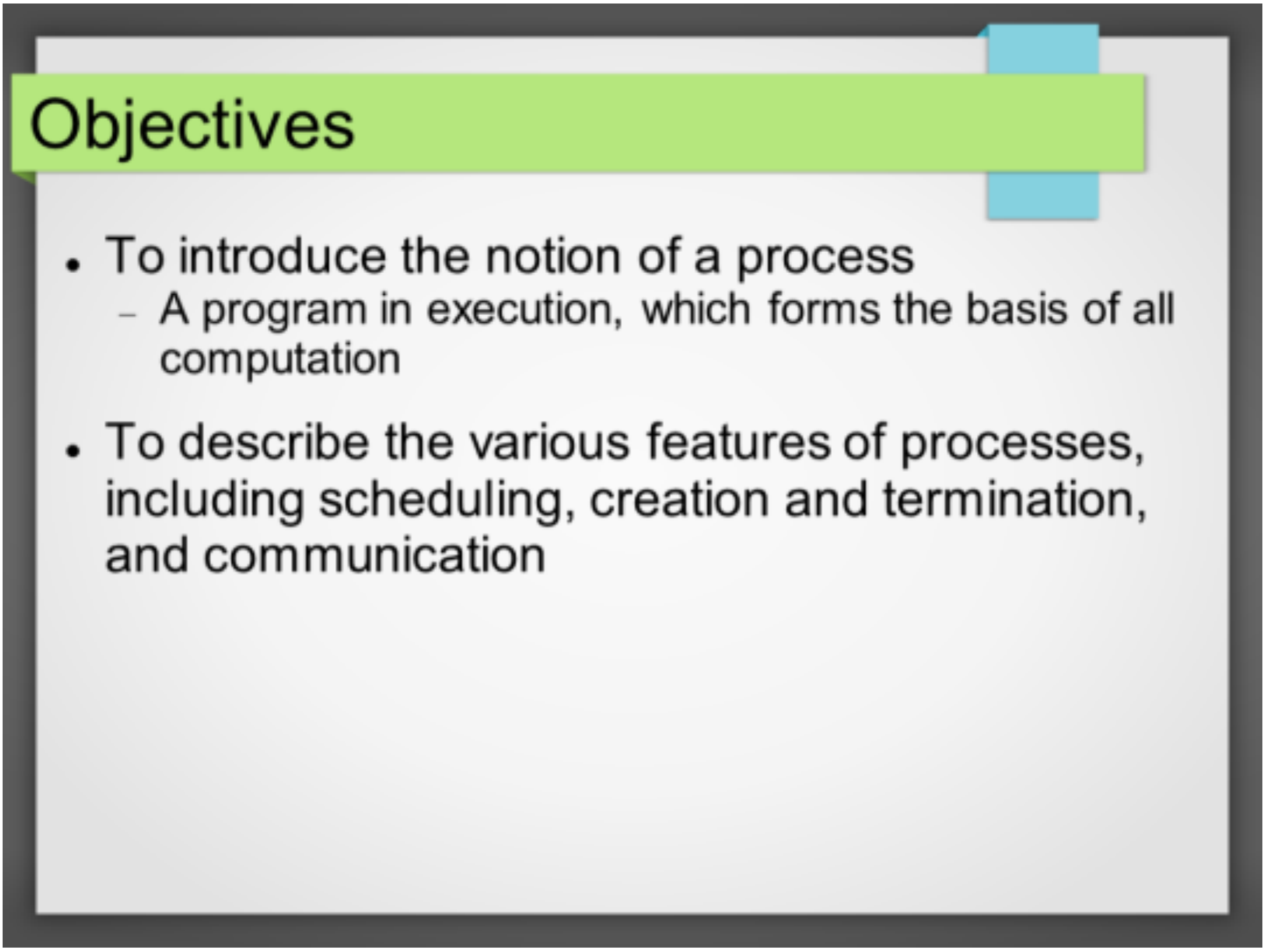
CST 347 ?? Real Time OS



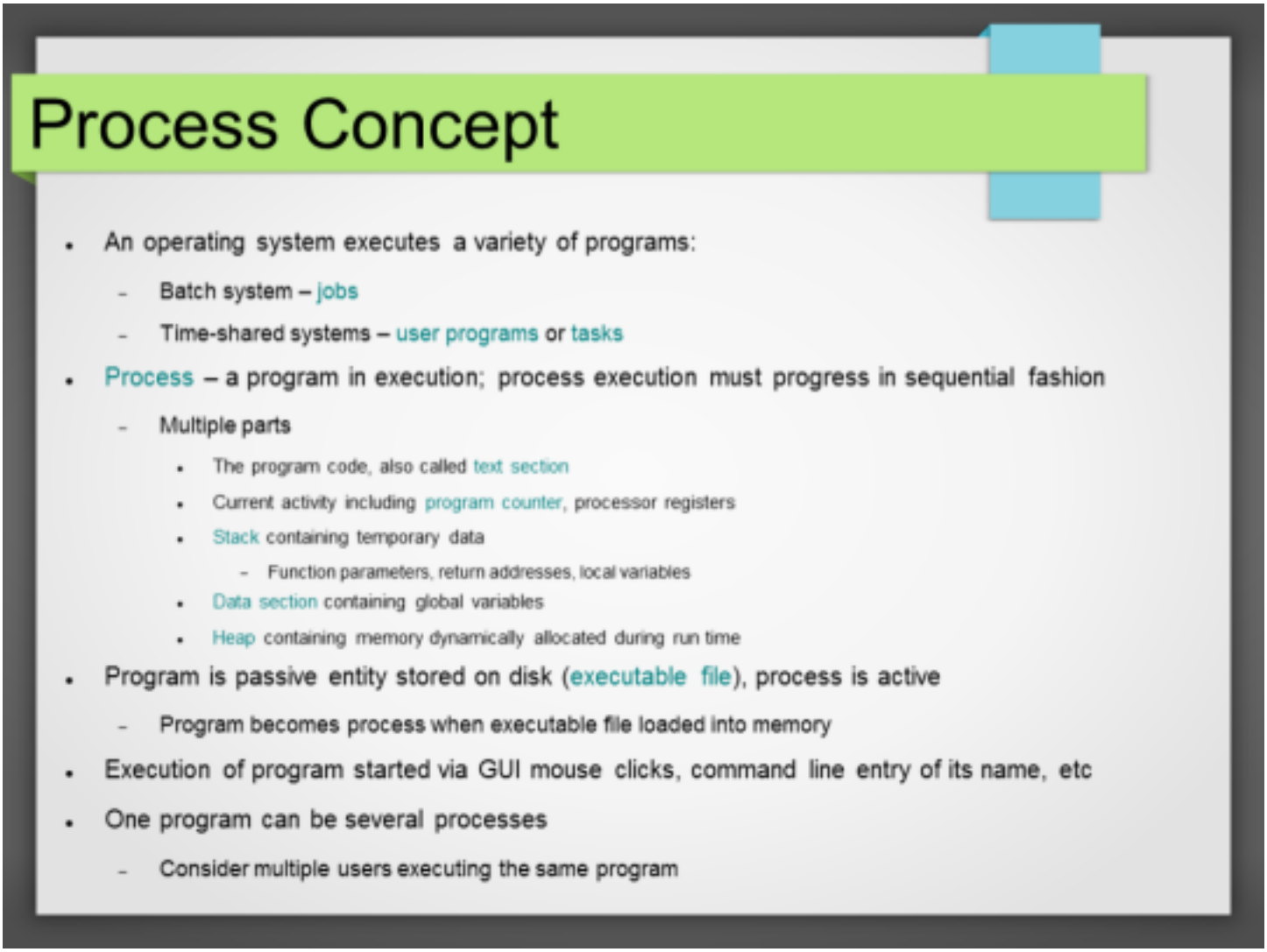
Definitions



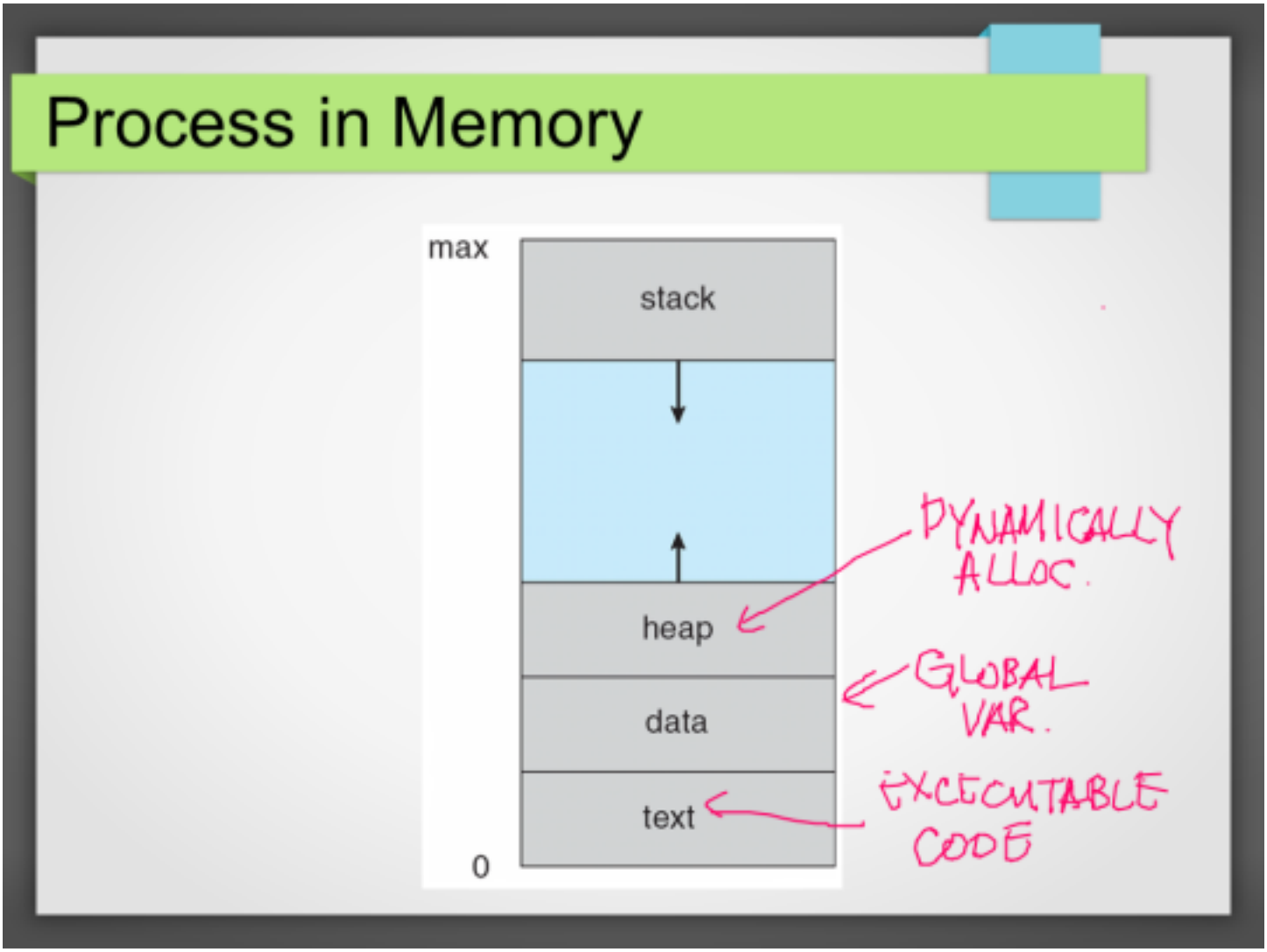
Objectives


Objectives
To introduce the notion of a process
A program in execution, which forms the basis of all computation
To describe the various features of processes, including scheduling, creation and termination, and communication

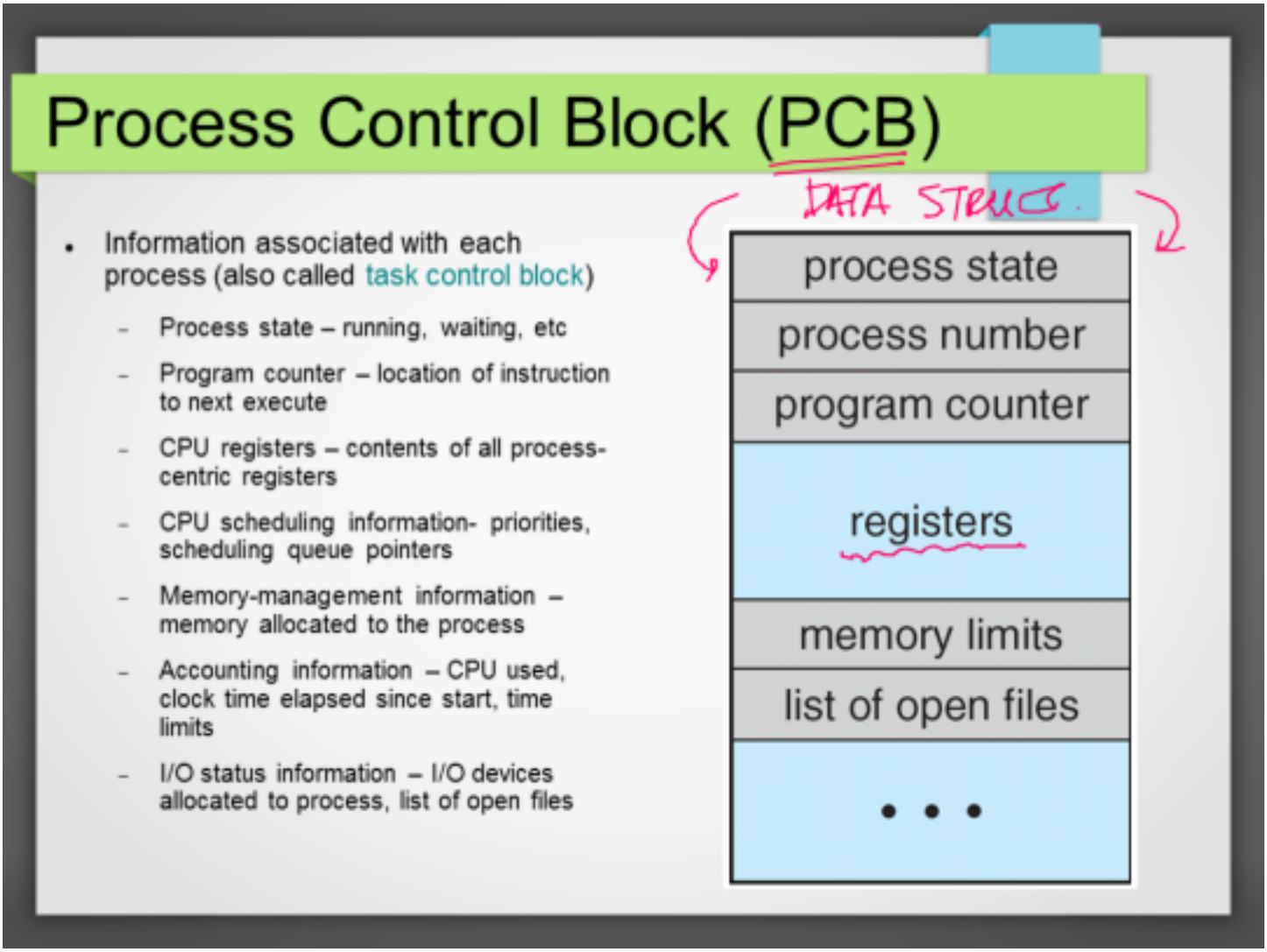
Process Concept



Process in Memory



Process Control Block (PCB)



Process State

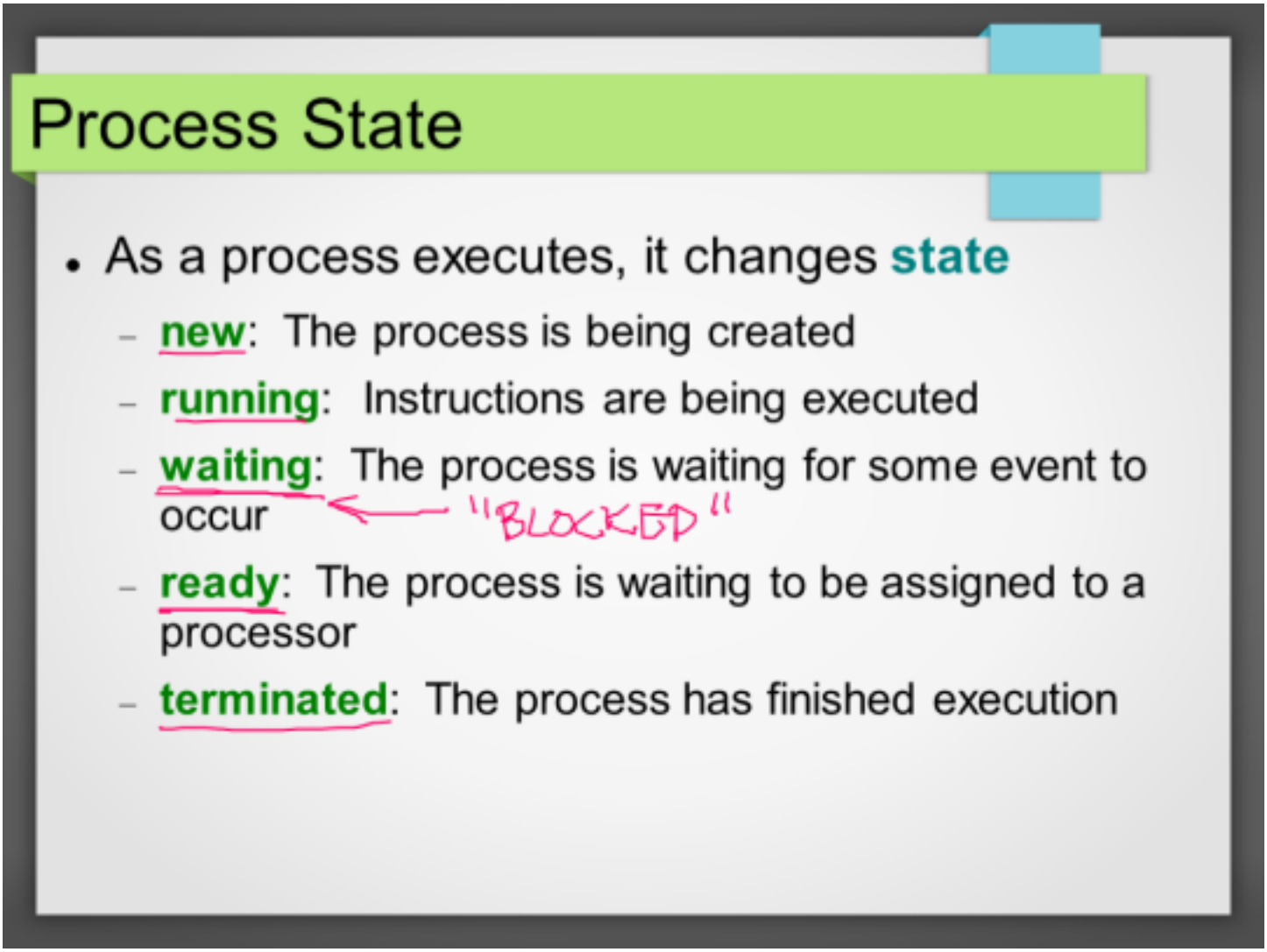
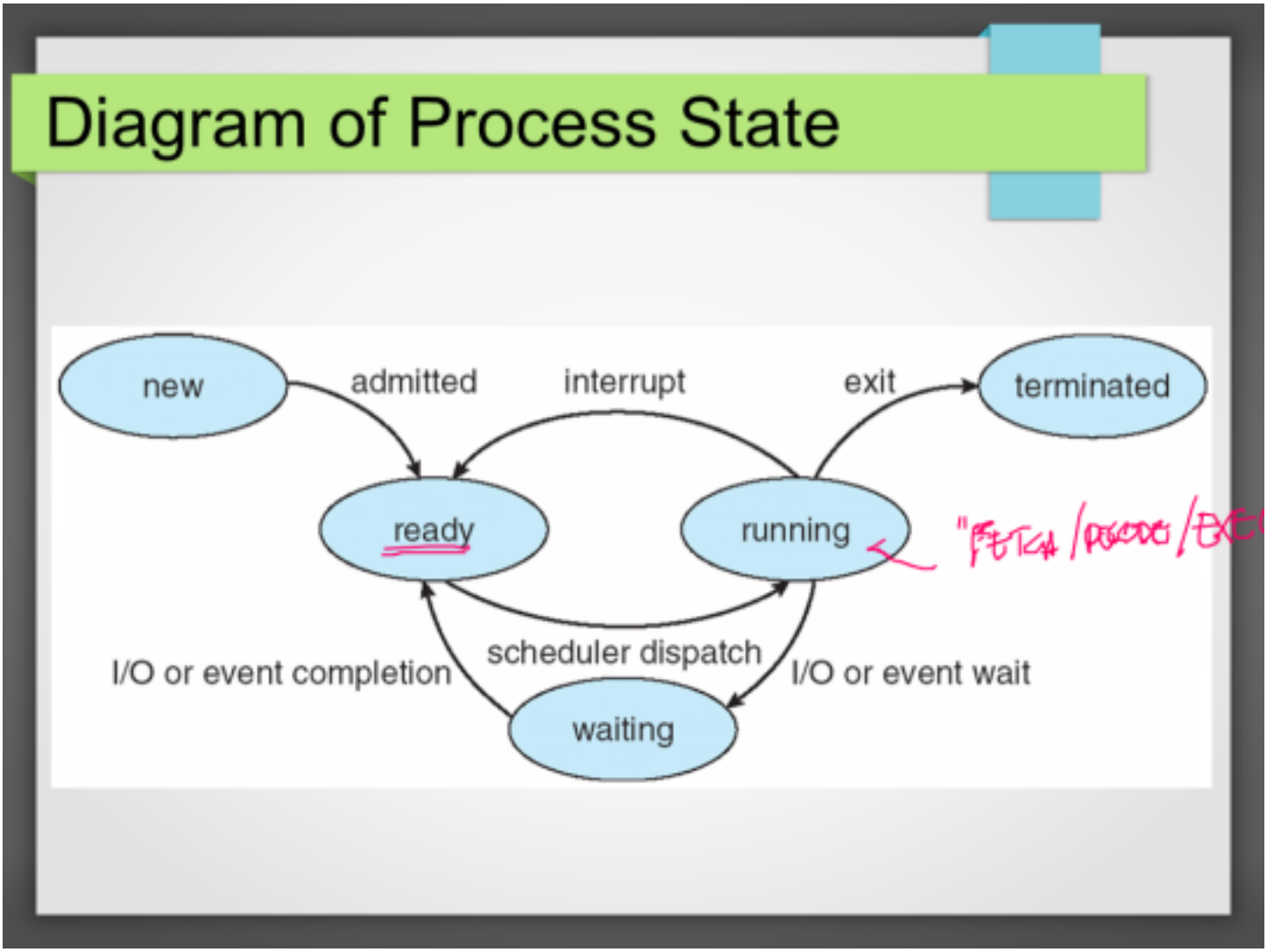


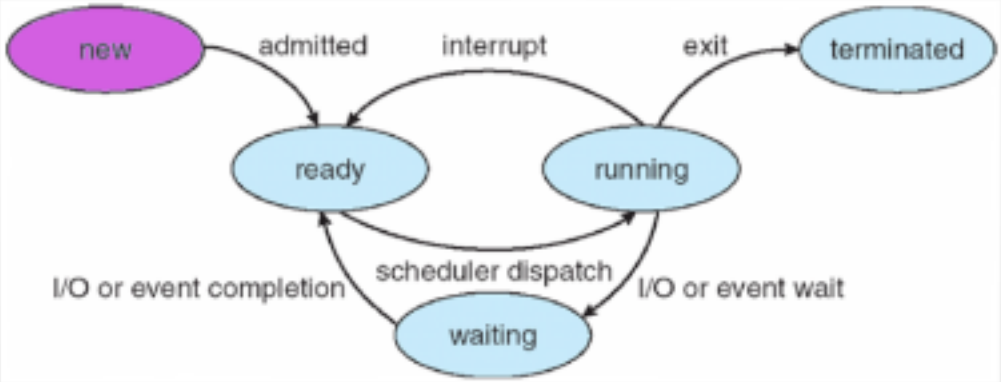
Diagram of Process State



Process State Transition

Process State Transition

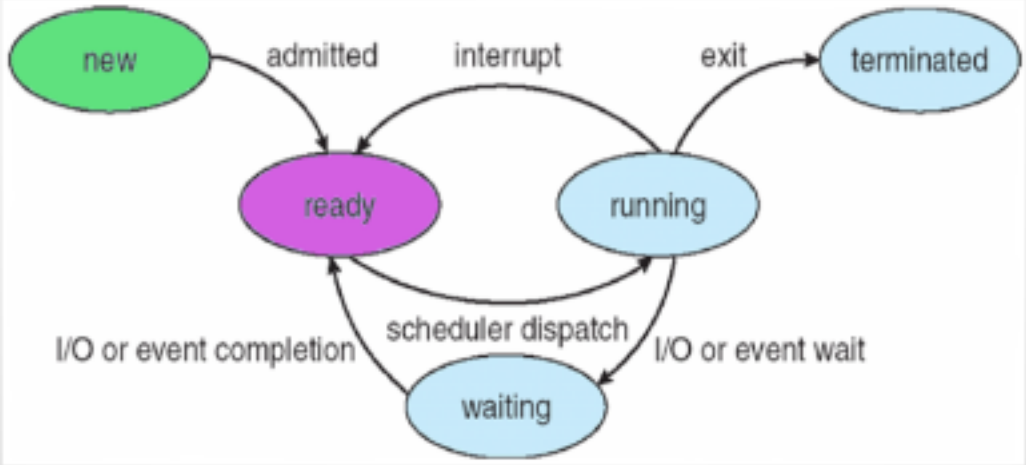
- Process Creation
  - Scheduler creates the Process Control Block and places it in the new list
    - Create process data segment.
    - Create process code segment.
    - Load op codes from disk into memory
    - Build run-time stack.



Process State Transition

Process State Transition

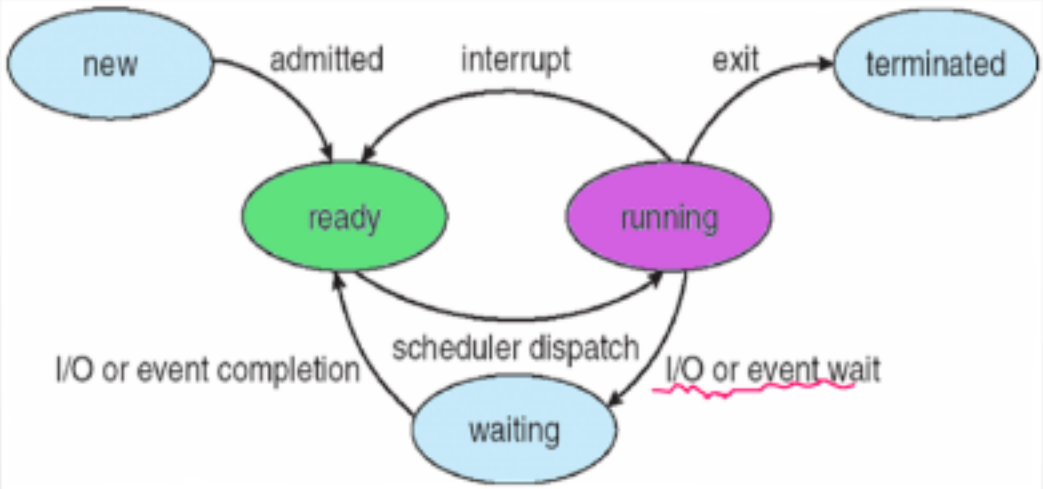
- Process is admitted by the system and goes to the ready state
  - Process system call is done and ready for execution
  - Process Control Block marked as active



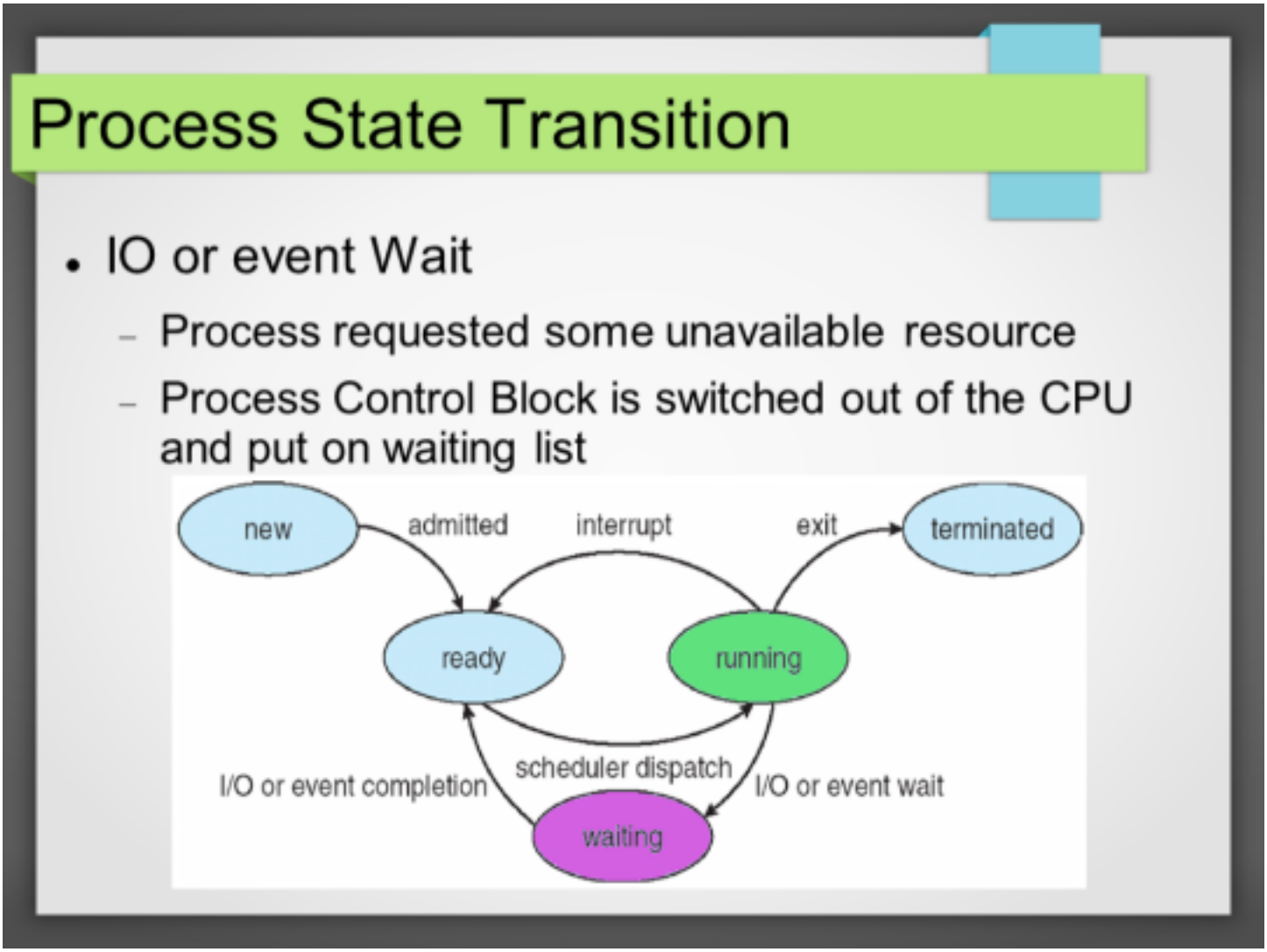
Process State Transition

Process State Transition

- Scheduler dispatch
  - Process Control Block is switched into the CPU based on Scheduling algorithm



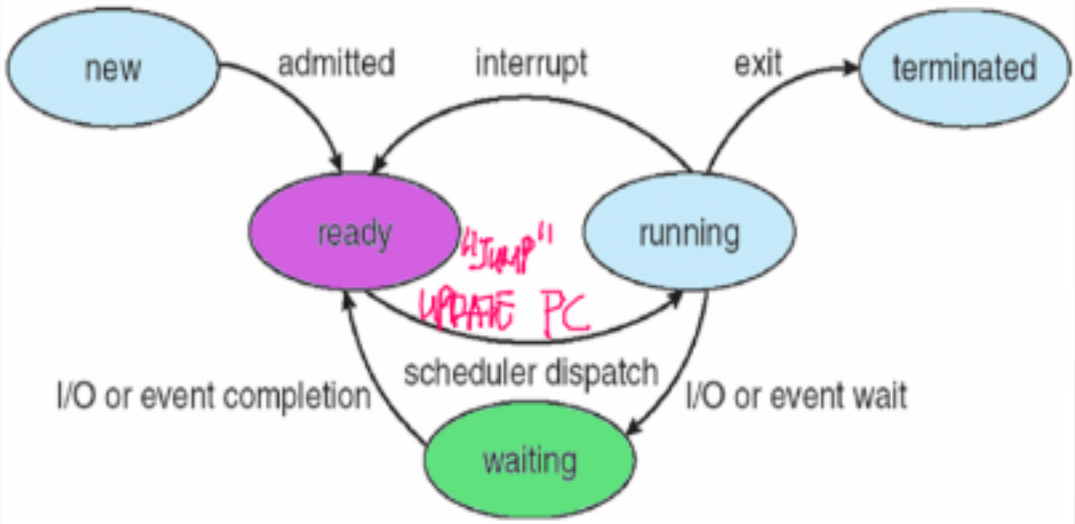
Process State Transition



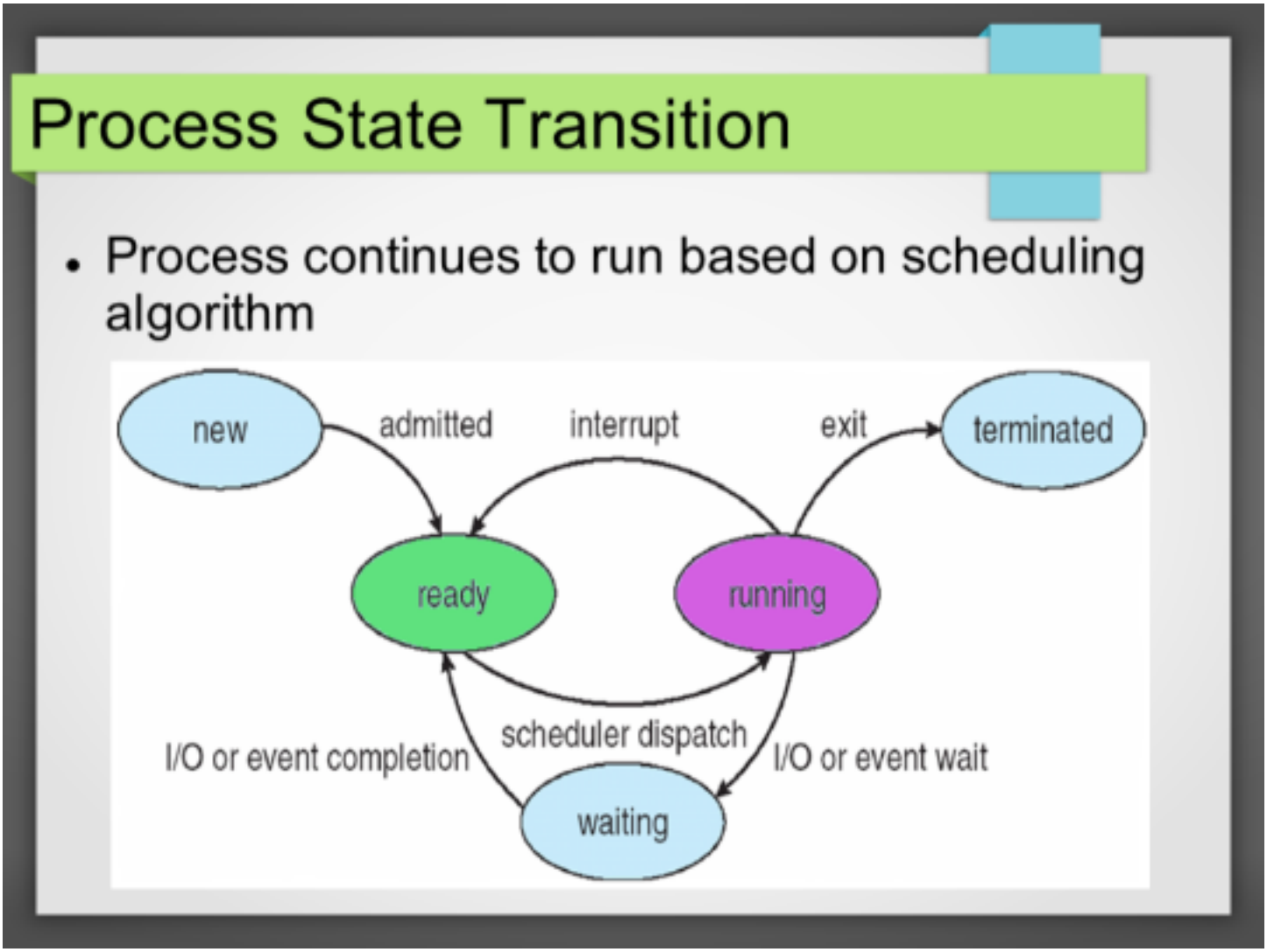
Process State Transition

Process State Transition

- Requested resource becomes available
  - Process Control blocked moved from waiting to ready (or active) state



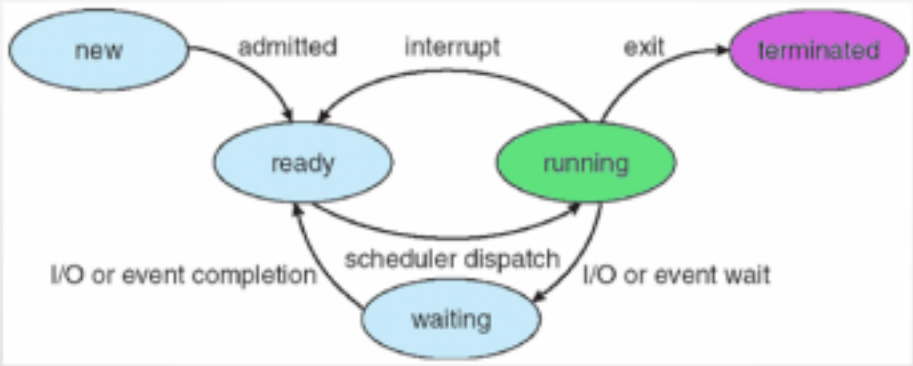
Process State Transition



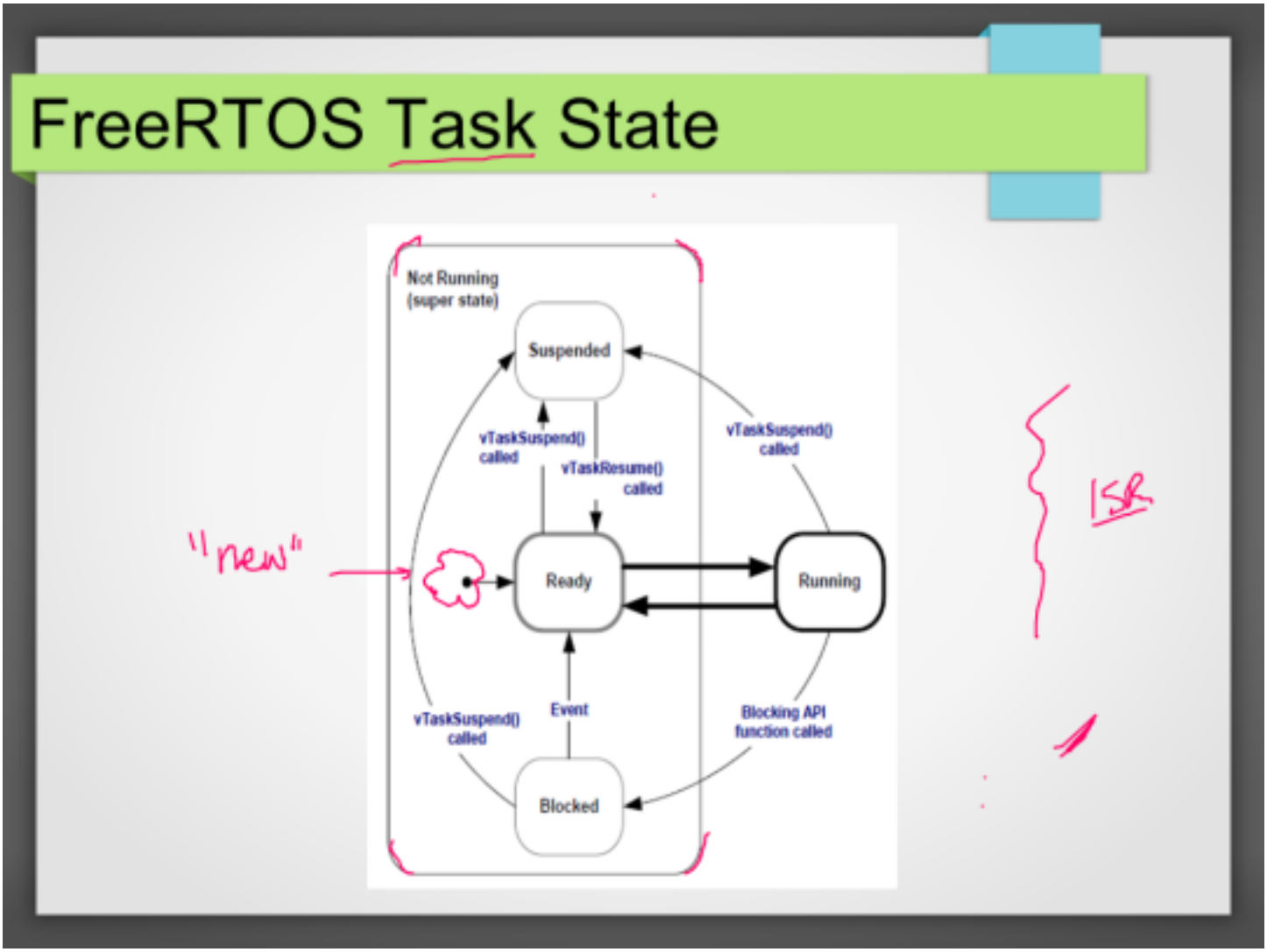
Process State Transition

Process State Transition

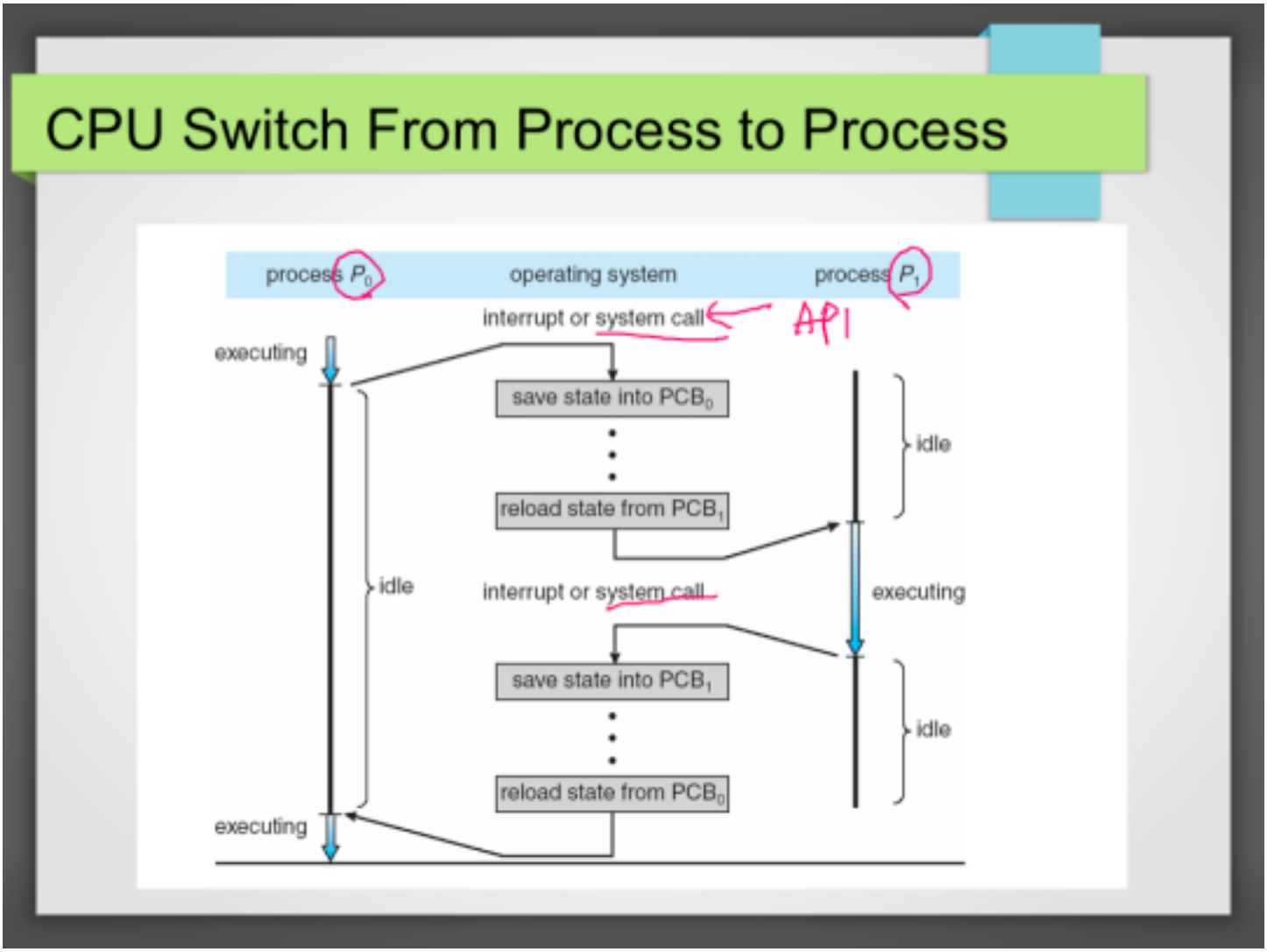
- Process Completes or is terminated by another process
  - Process Resources are returned to OS
  - Process Memory Segment is cleaned up
  - Process Runtime stack is cleaned up



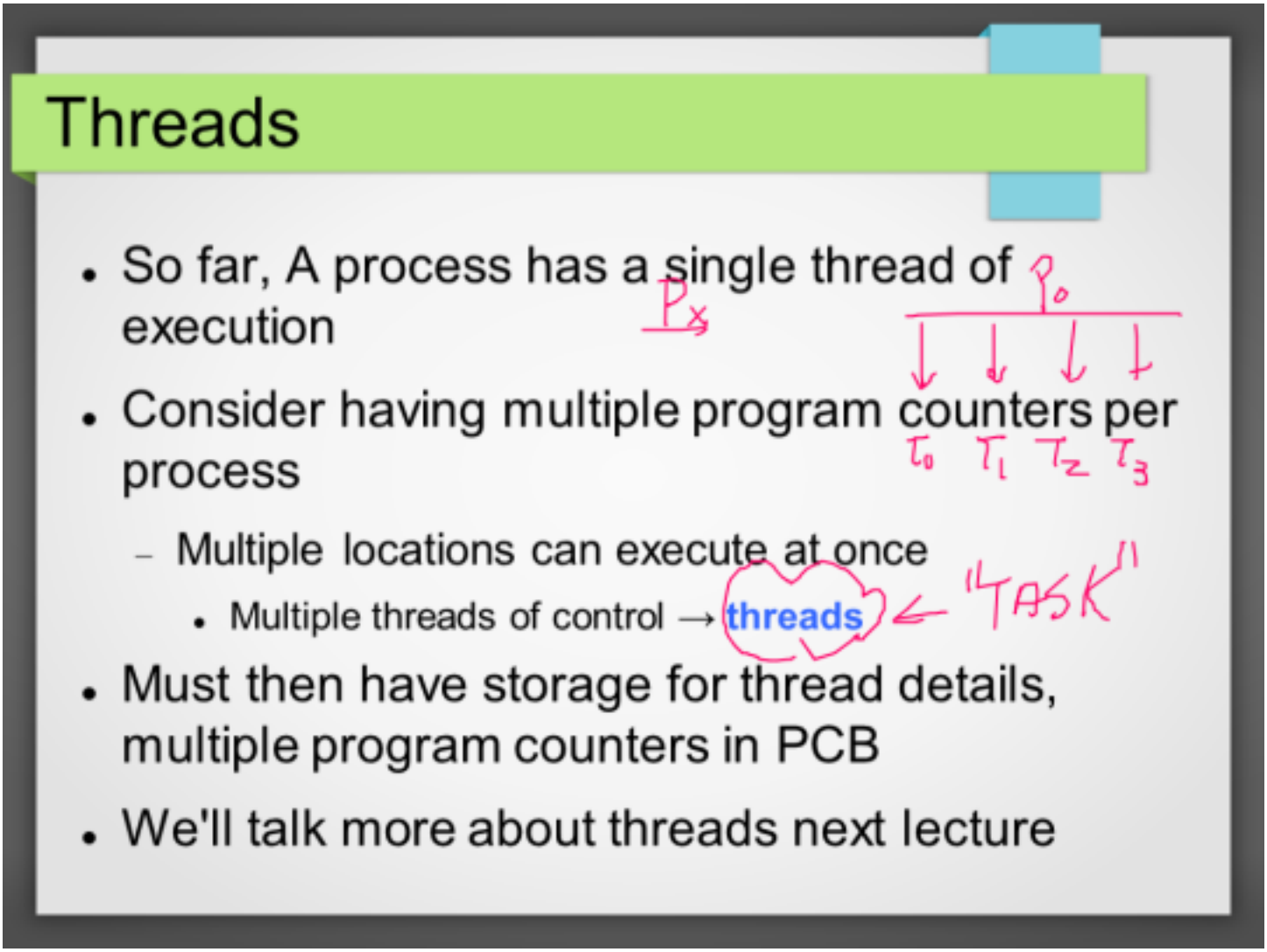
FreeRTOS Task State



CPU Switch From Process to Process



Threads


Threads
So far, A process has a single thread of execution
Consider having multiple program counters per process
Multiple locations can execute at once
Multiple threads of control → threads
Must then have storage for thread details, multiple program counters in PCB
We'll talk more about threads next lecture

Day 04 Ended here.

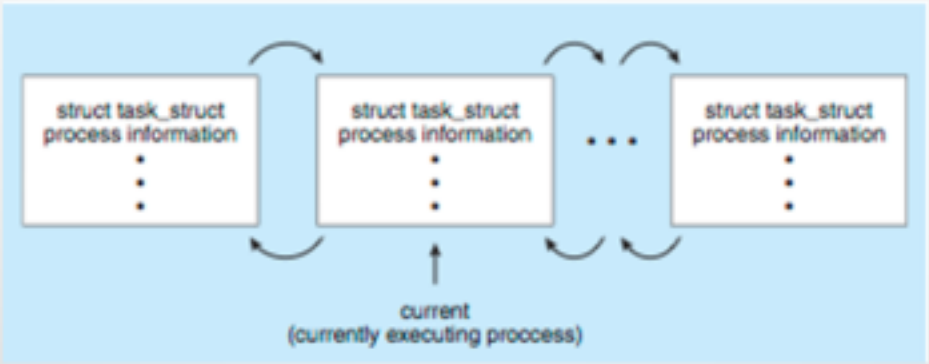
Will continue after this point.

Process Representation in Linux

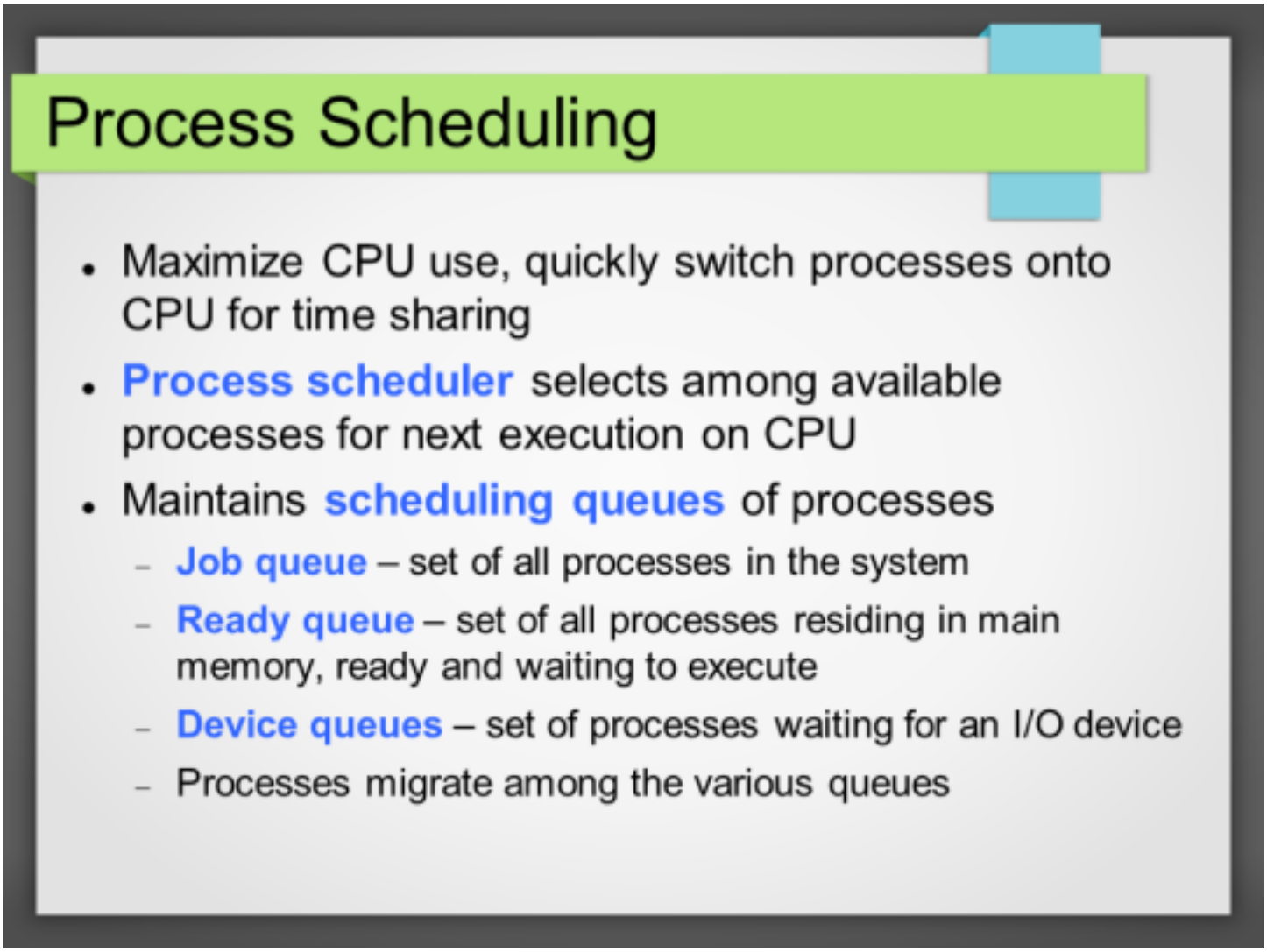
Process Representation in Linux

- Represented by the C structure

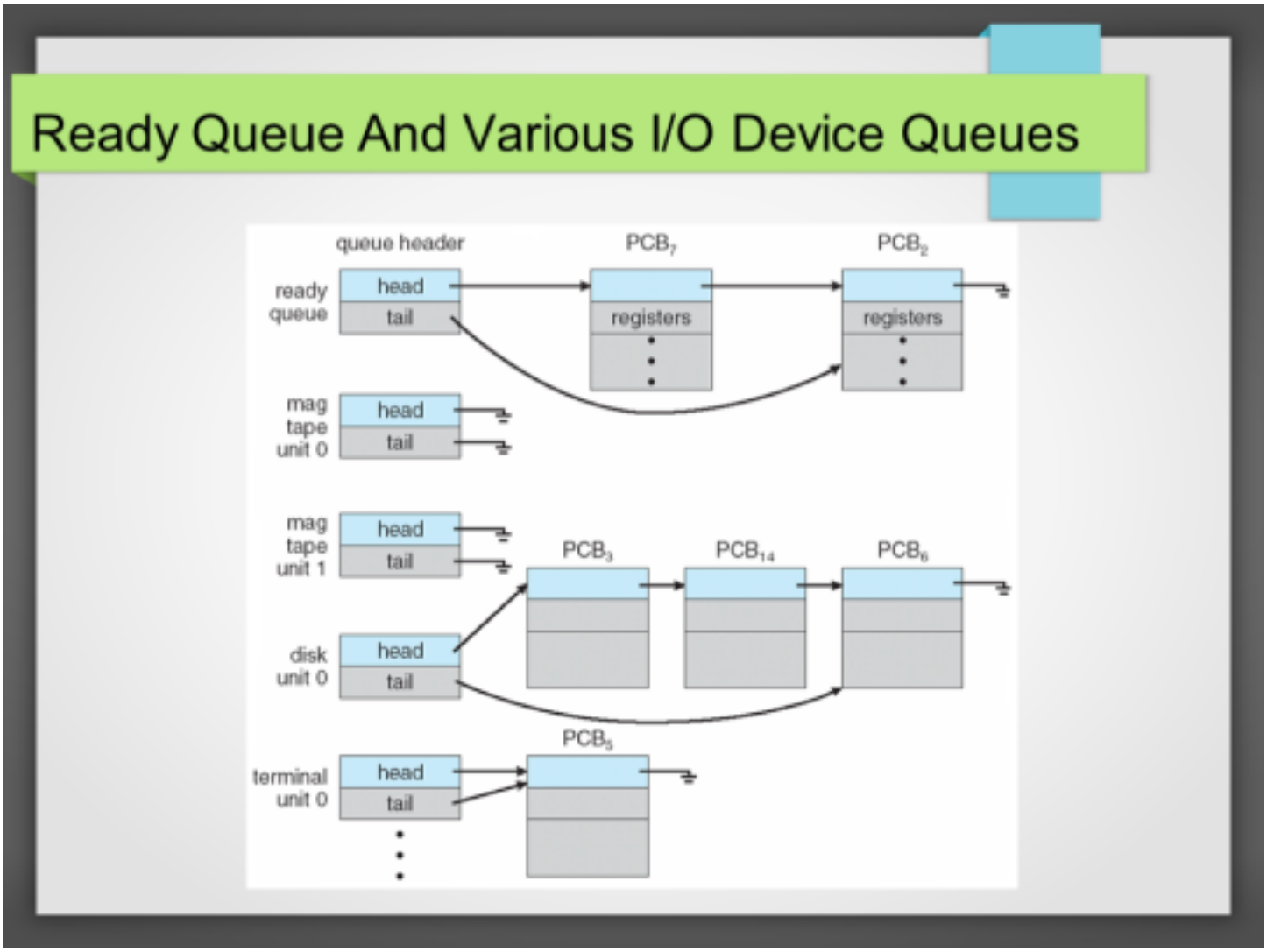
```
struct task_struct
{  pid_t pid;                    /* process identifier */
     long state;                 /* state of the process */
unsigned int time_slice;         /* scheduling information */
struct task_struct *parent;      /* this process's parent */
struct list_head children;       /* this process's children */
struct files_struct *files;      /* list of open files */
struct mm_struct *mm;            /* address space of this process */
}
```



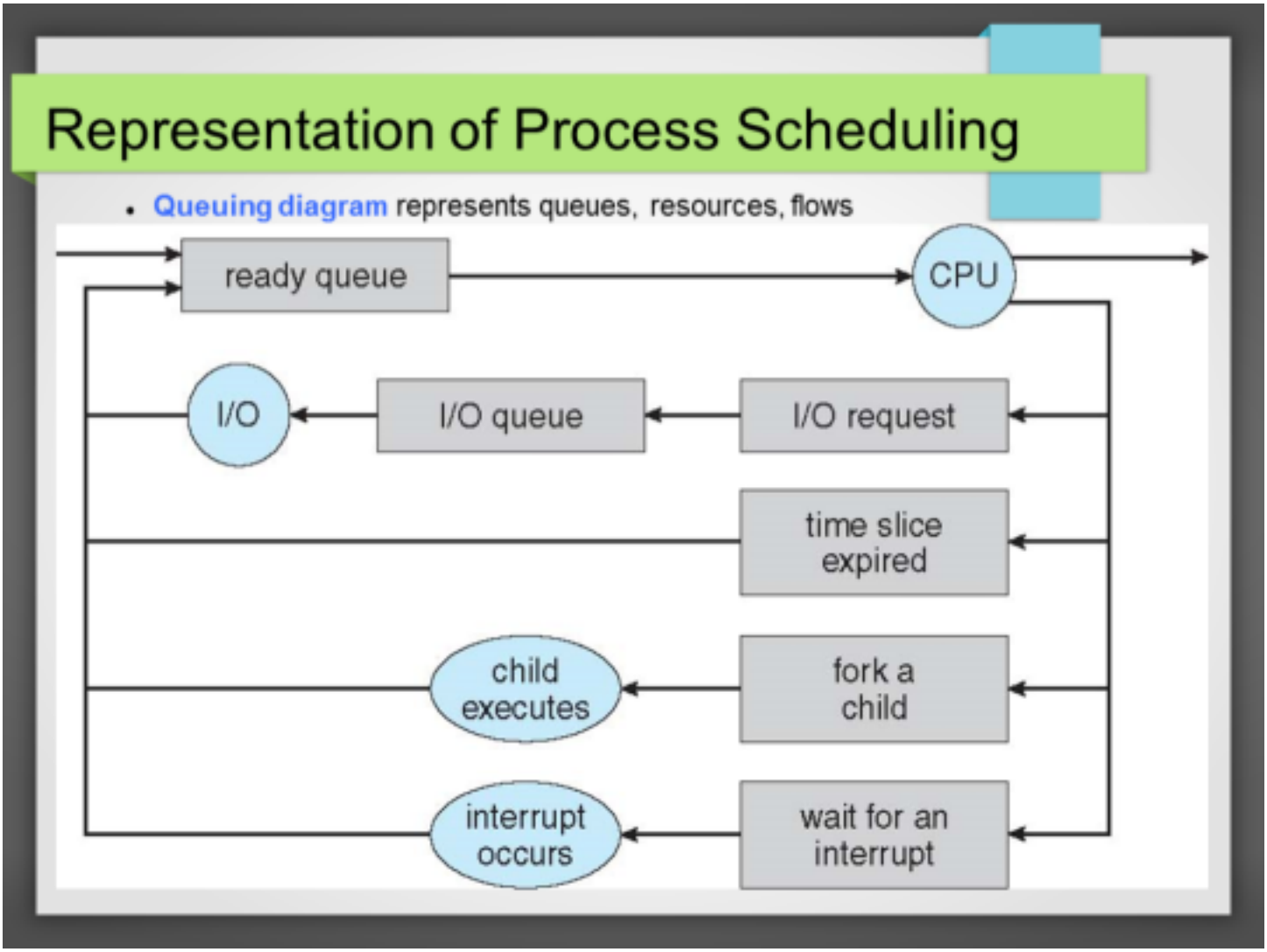
Process Scheduling



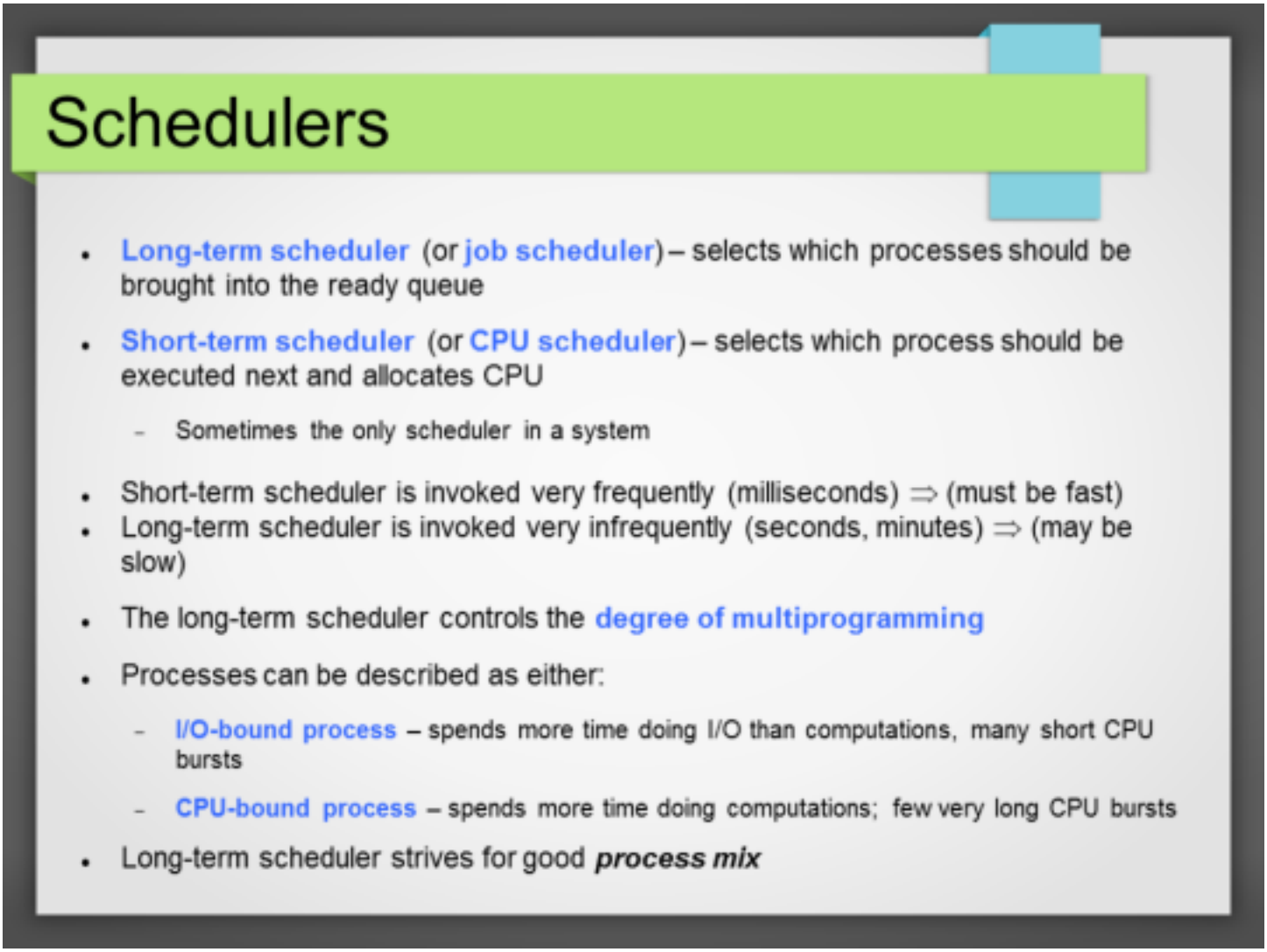
Ready Queue And Various I/O Device Queues



Representation of Process Scheduling



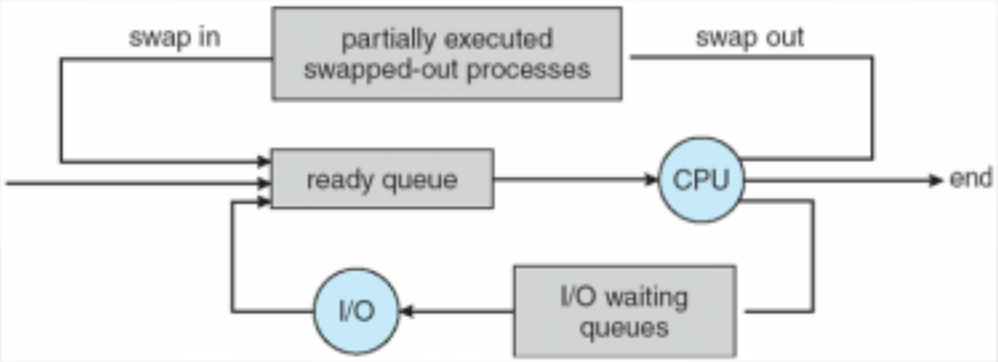
Schedulers


Schedulers
Long-term scheduler (or job scheduler) – selects which processes should be brought into the ready queue
Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU
Sometimes the only scheduler in a system
Short-term scheduler is invoked very frequently (milliseconds) ⇒ (must be fast)
Long-term scheduler is invoked very infrequently (seconds, minutes) ⇒ (may be slow)
The long-term scheduler controls the degree of multiprogramming
Processes can be described as either:
I/O-bound process – spends more time doing I/O than computations, many short CPU bursts
CPU-bound process – spends more time doing computations; few very long CPU bursts
Long-term scheduler strives for good process mix

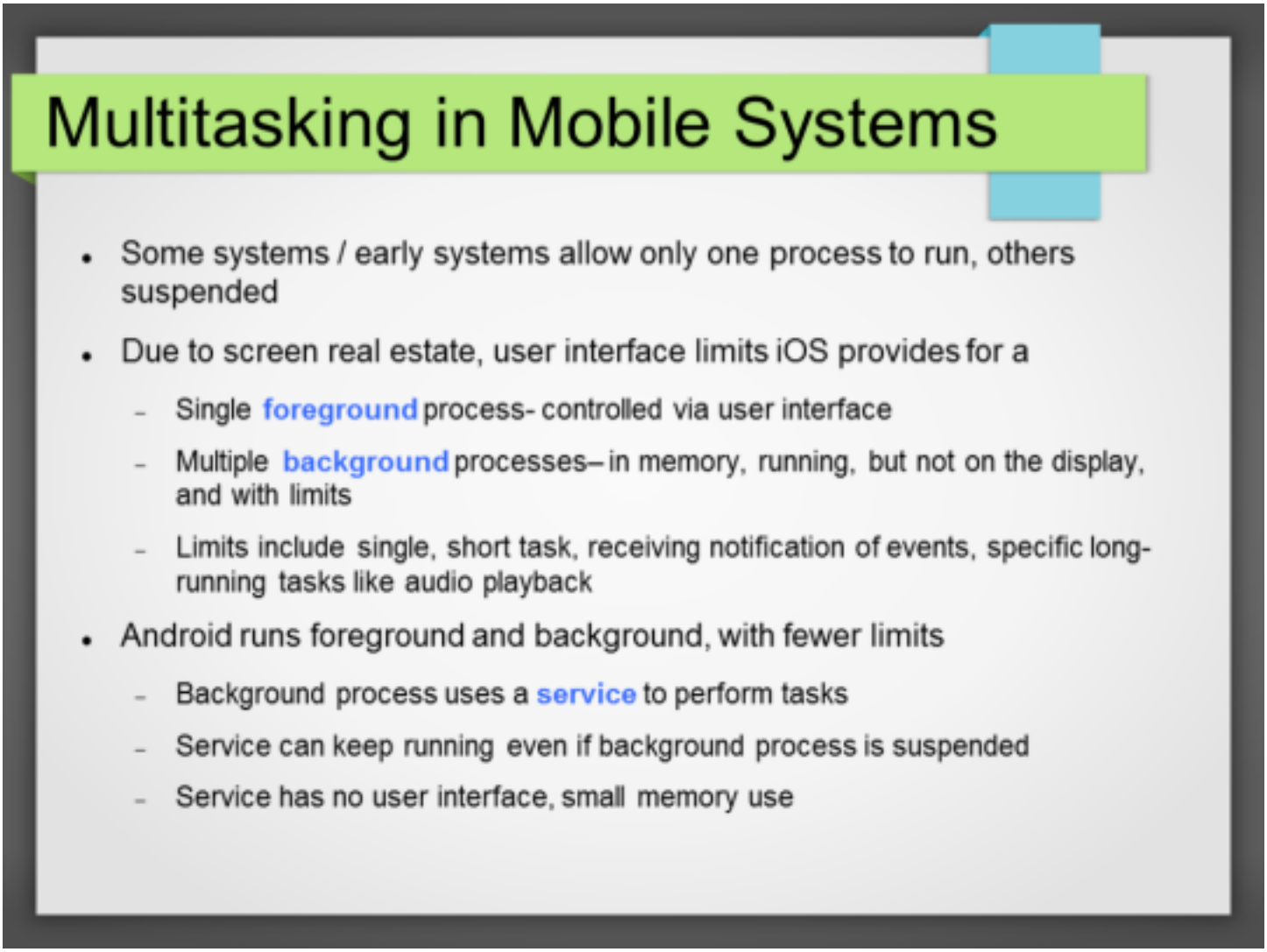
Addition of Medium Term Scheduling

Addition of Medium Term Scheduling

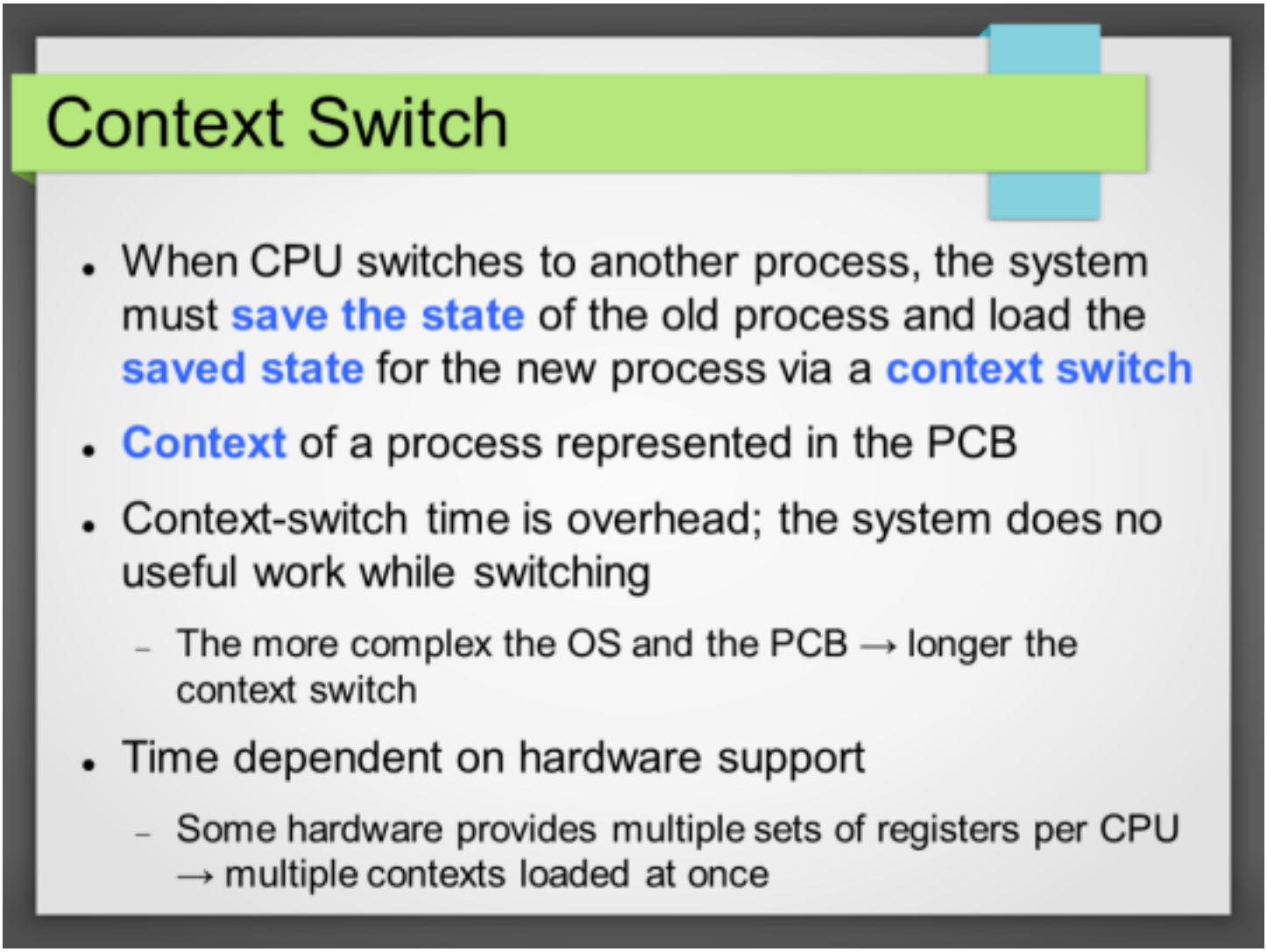
- Medium-term scheduler can be added if degree of multiple programming needs to decrease
  - Remove process from memory, store on disk, bring back in from disk to continue execution: swapping



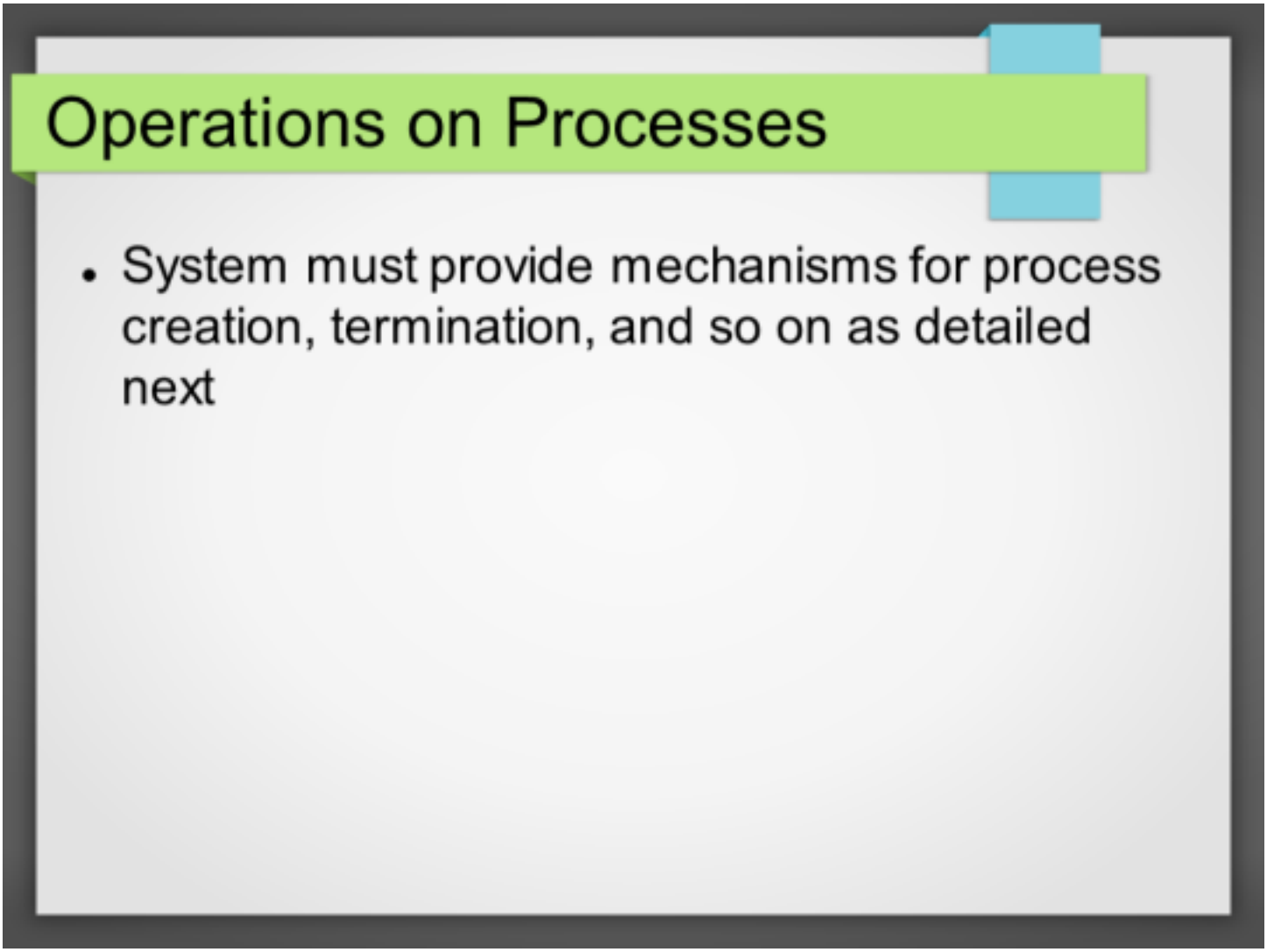
Multitasking in Mobile Systems



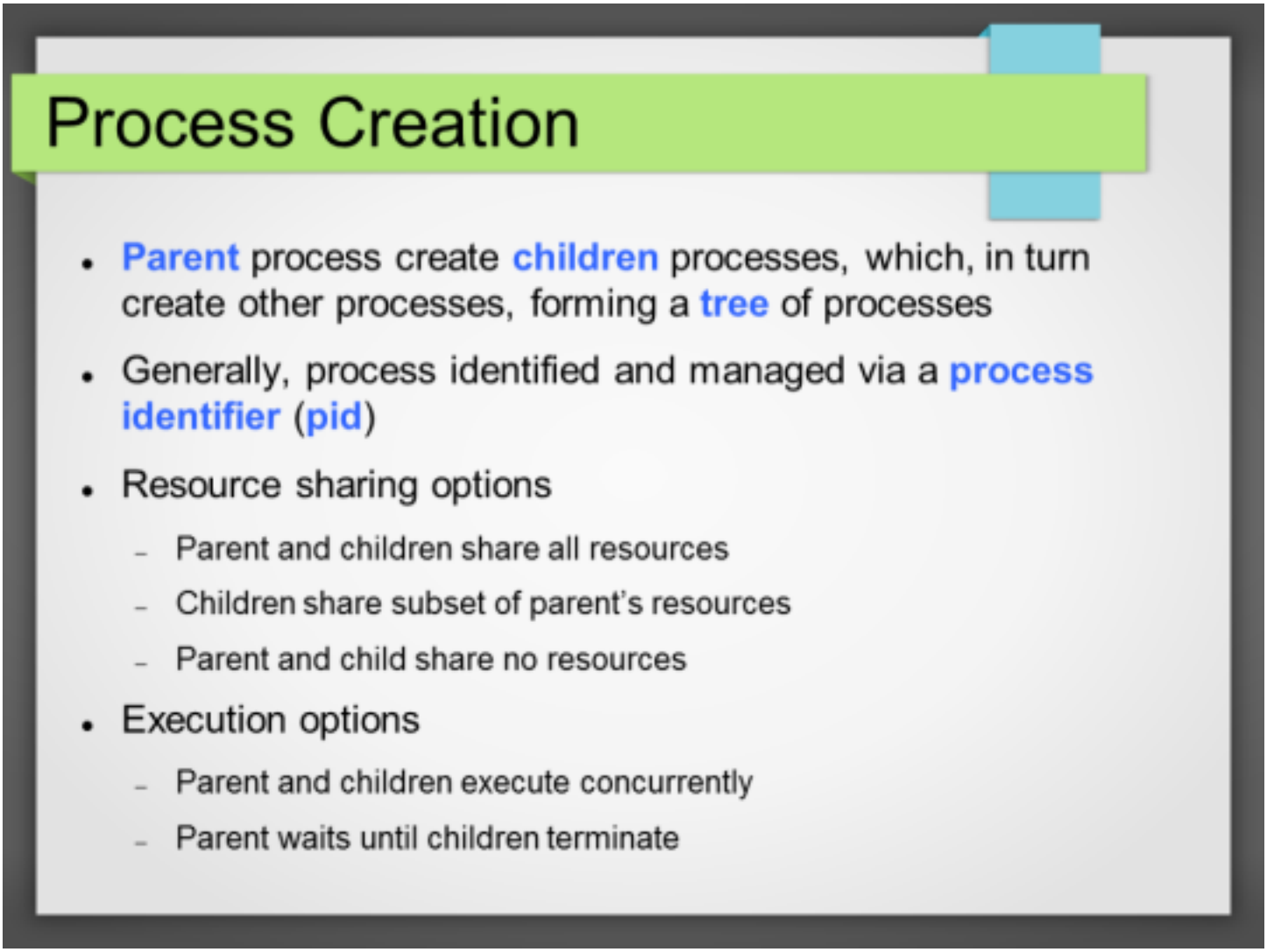
Context Switch



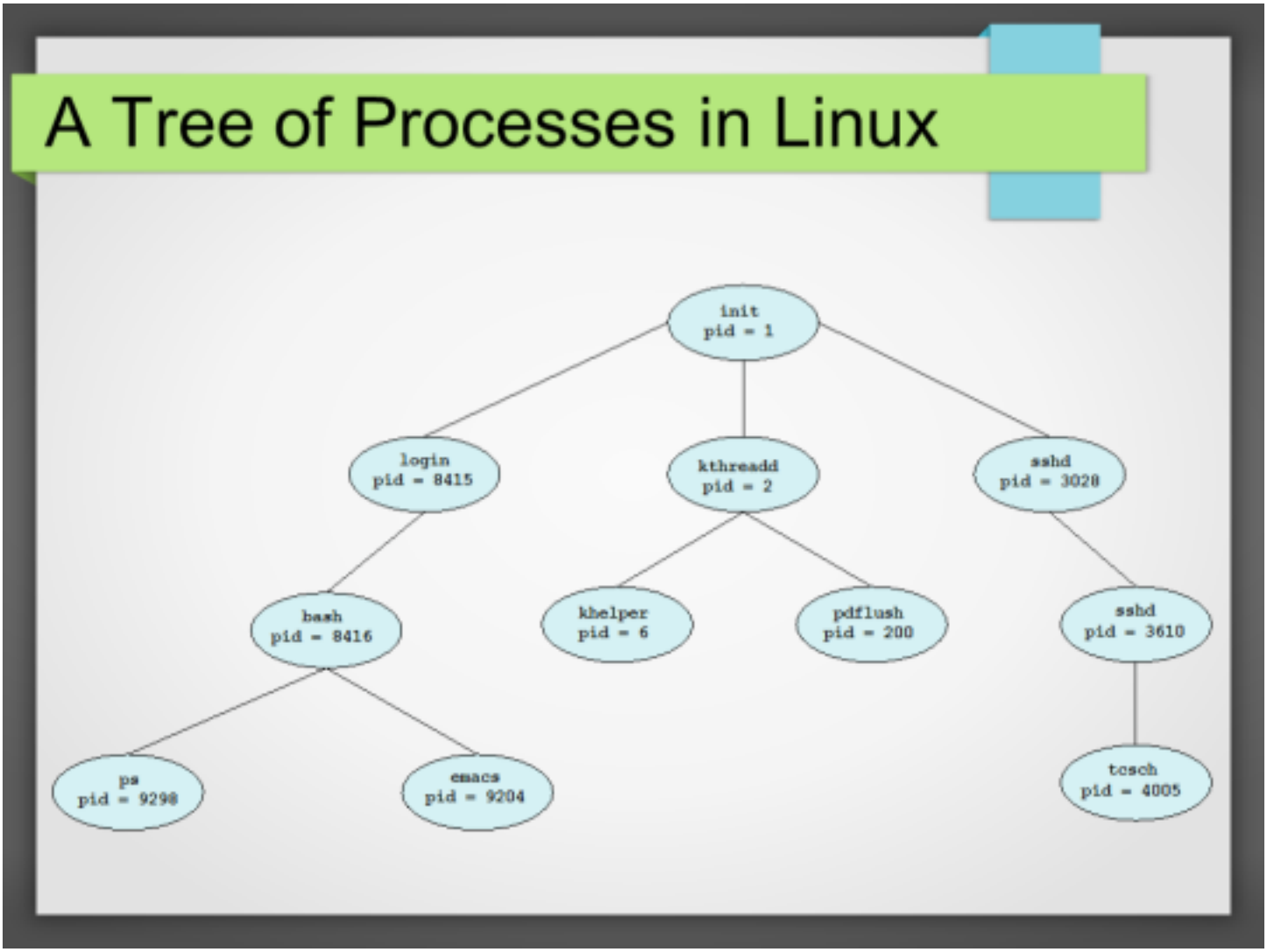
Operations on Processes



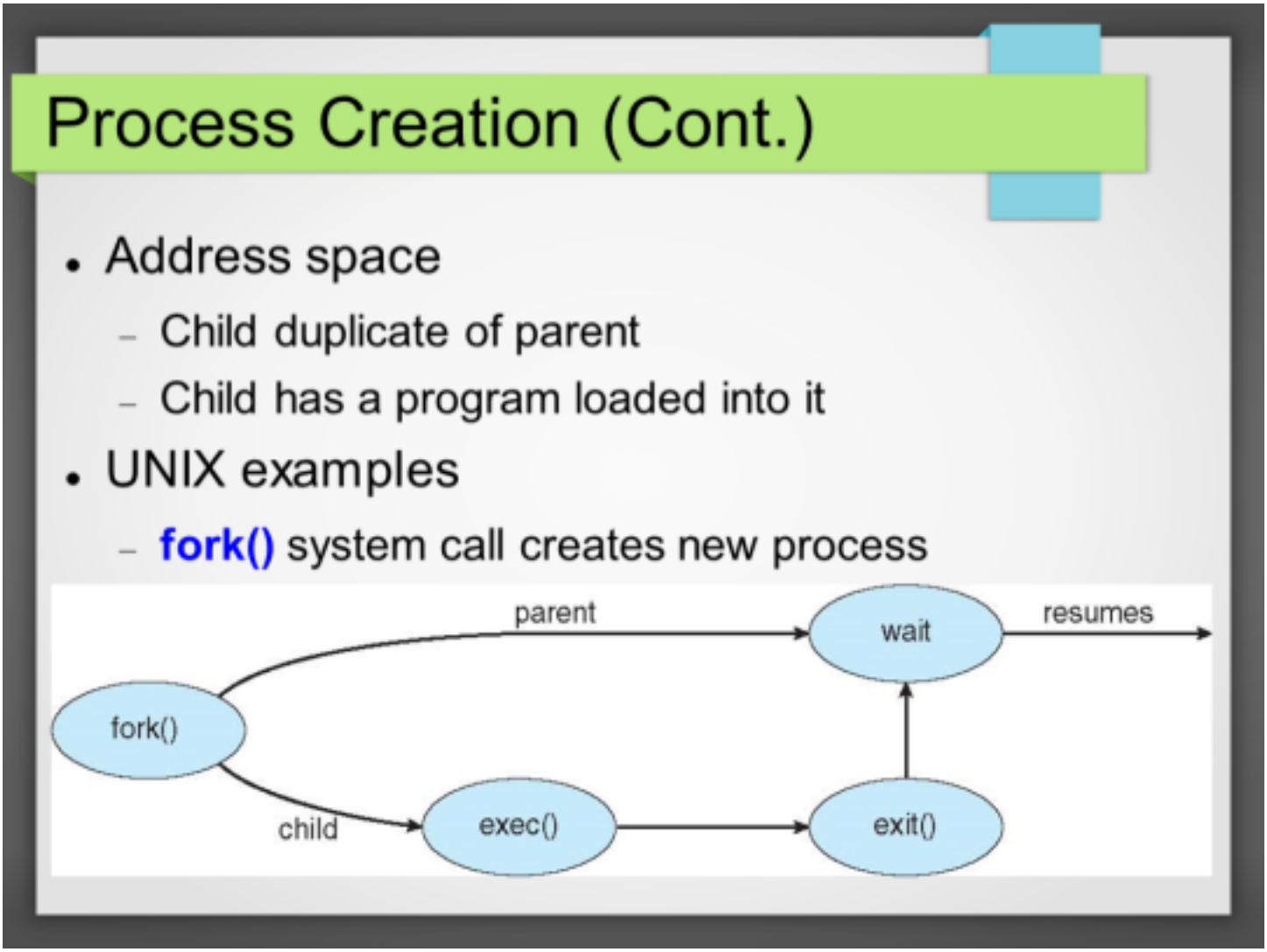
Process Creation



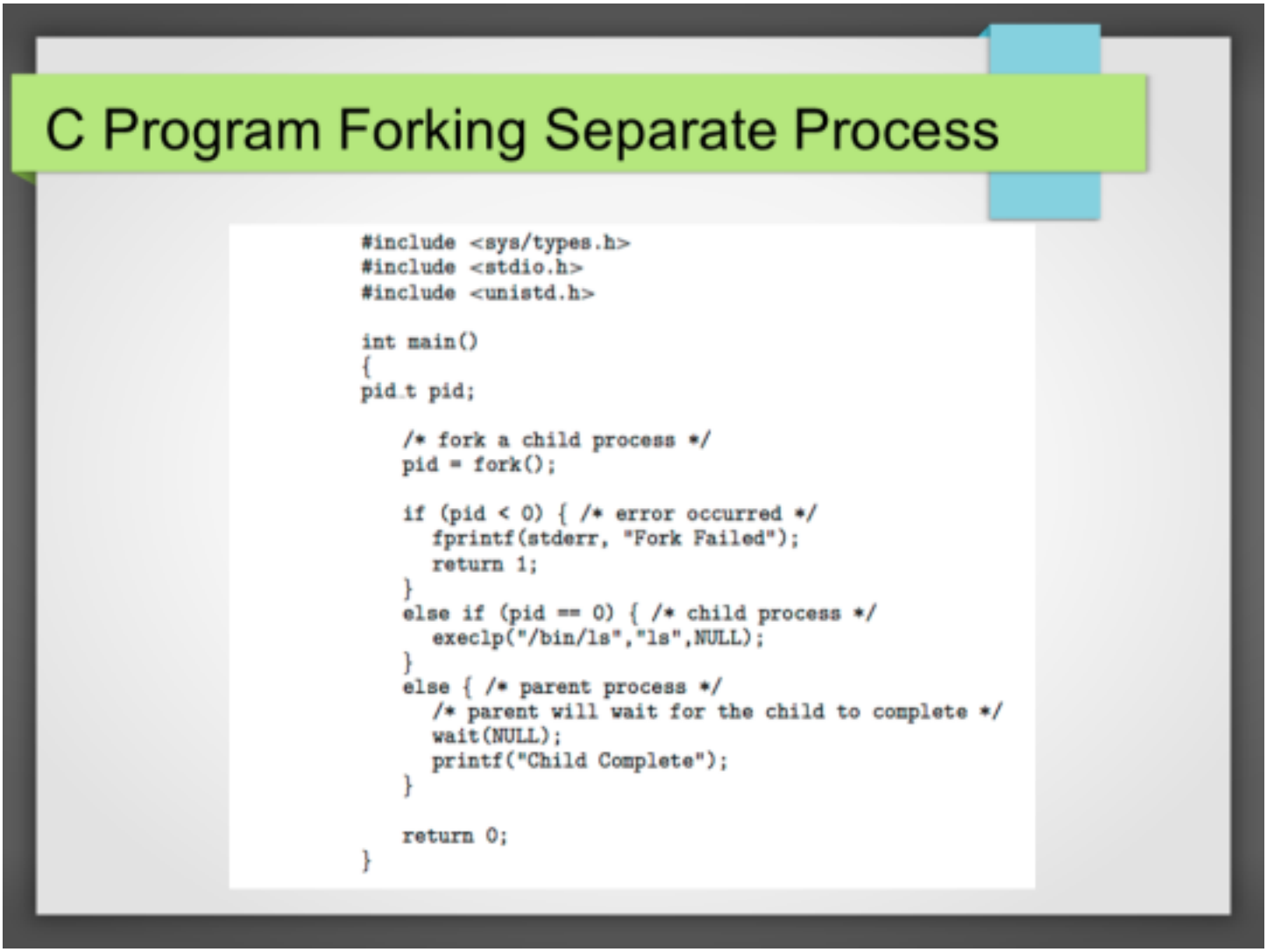
A Tree of Processes in Linux



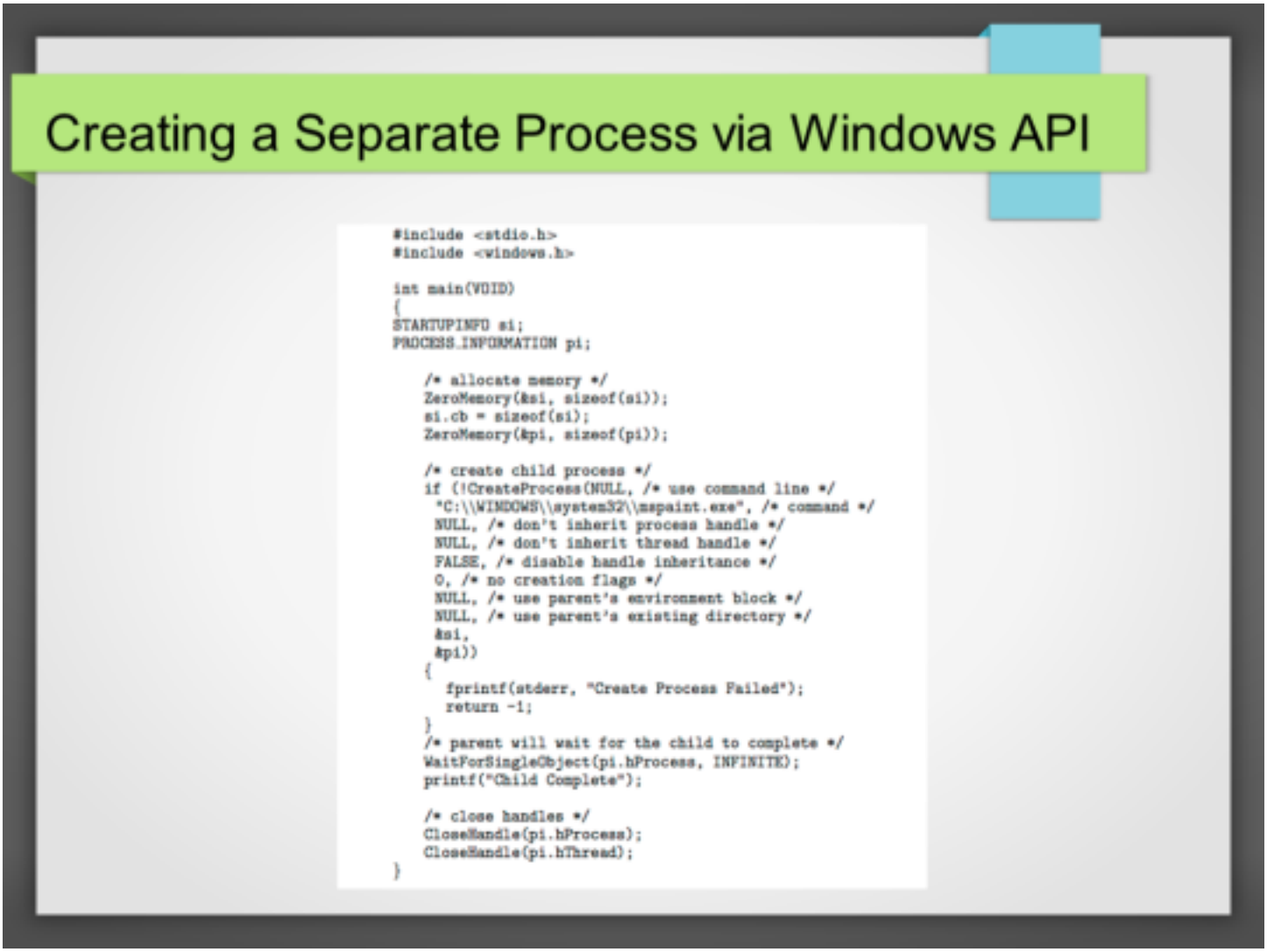
Process Creation (Cont.)



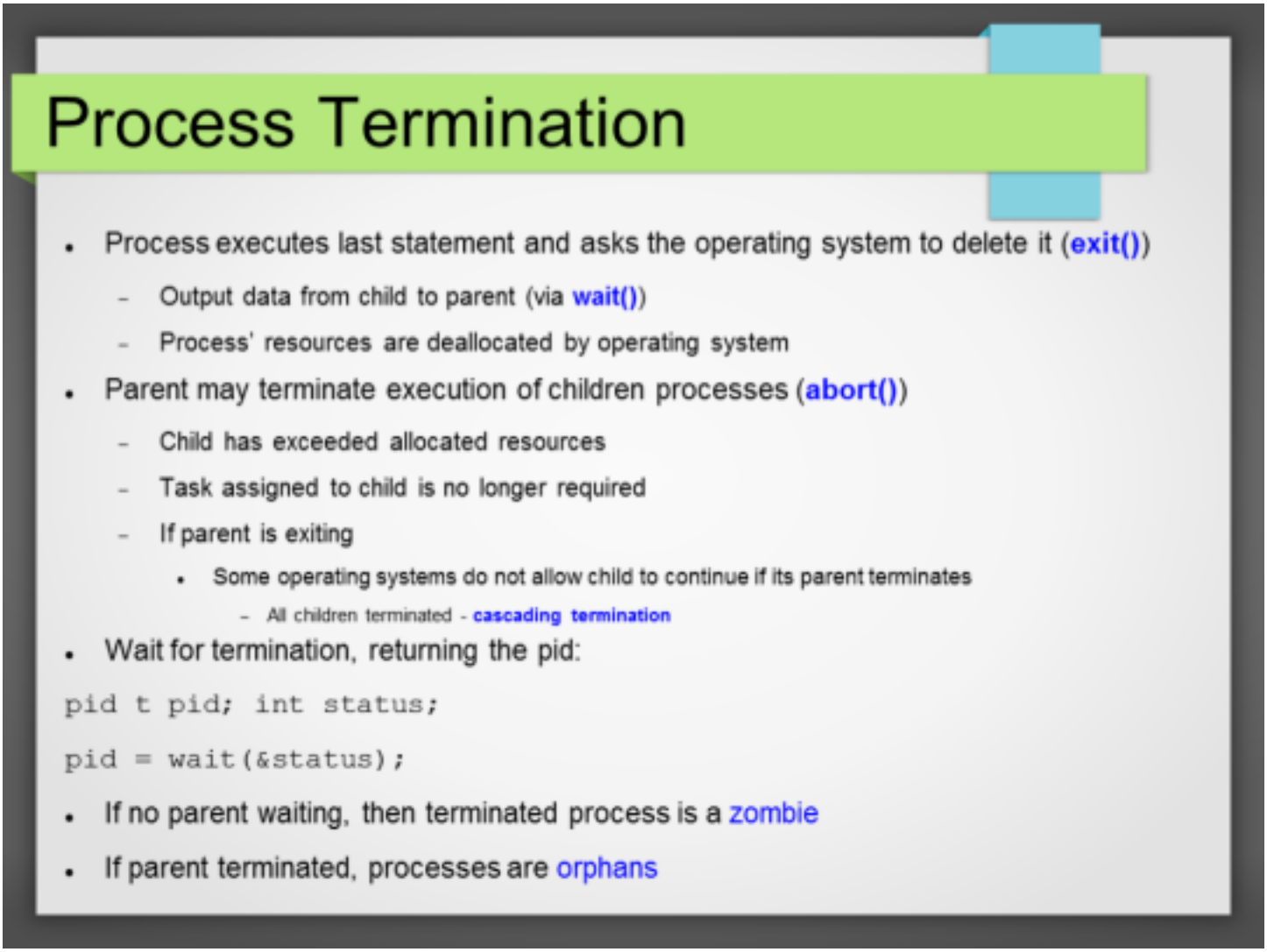
C Program Forking Separate Process



Creating a Separate Process via Windows API



Process Termination



Multiprocess Architecture ?? Chrome Browser