

Processes

Bringing It All Together

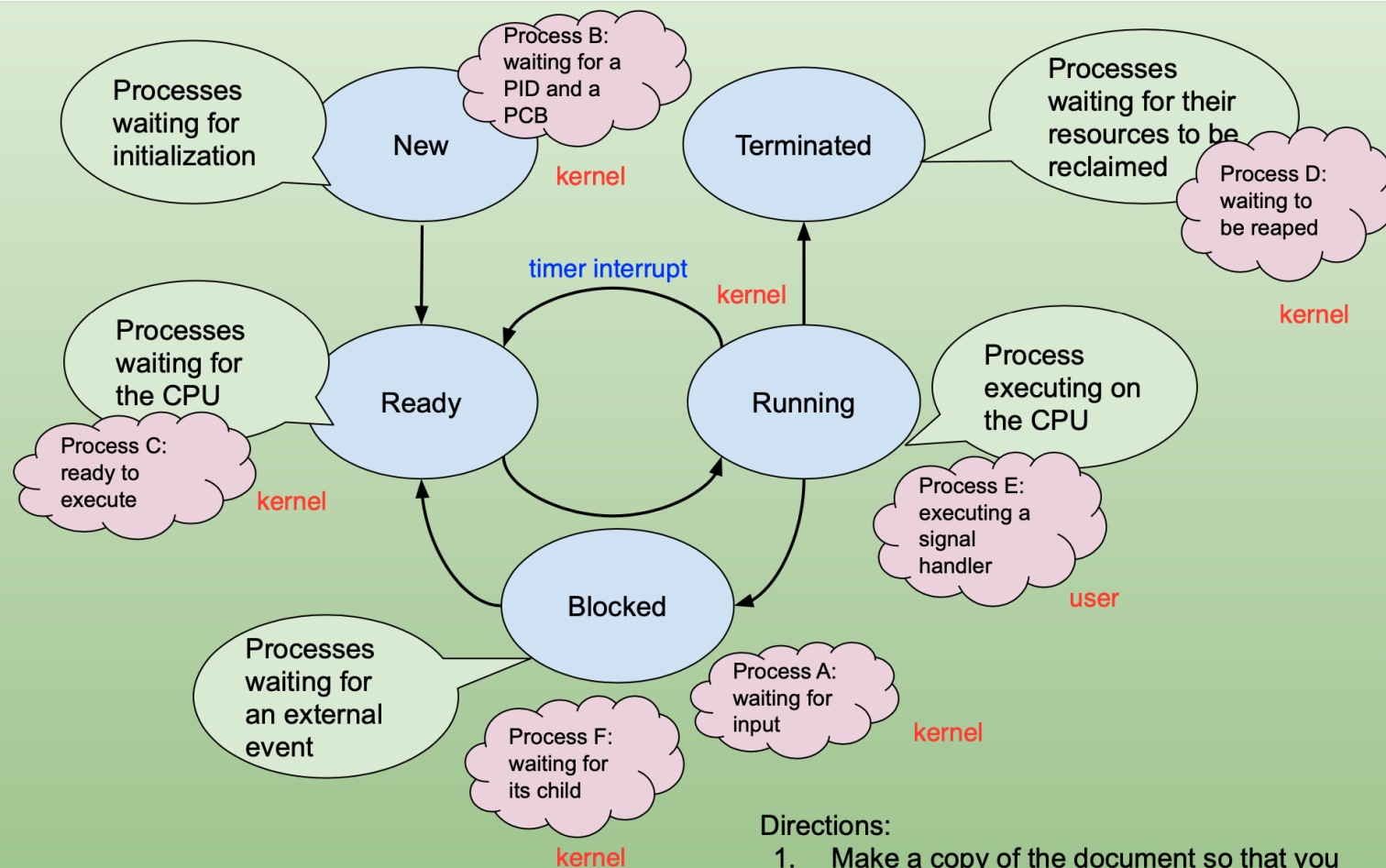
- A process is a unit of execution
 - Defines an address space
 - An abstraction for protection
 - Basic unit of execution in the OS
 - Processes are represented as Process Control Blocks in the OS
 - At any time, a process is either New, Ready, Blocked, Running, or Terminated
 - Processes are created and managed through system calls
 - system calls exist for other things, too
 - Supports the OS in its roles as referee and illusionist
- We're in a multiprogramming environment
- We also want to support timesharing and interactivity
- We must provide protection---dual-mode execution helps with that

Vocabulary

doesn't return	fork	child	pid_t
referee	heap segment	system calls	HAL
reaped	API	process status register	access is unrestricted
dual-mode execution	exceptions	running	kill
program	ready	address I/O directly	set of executable instructions
returns twice	CPU	halt the machine	orphan
shells	blocked	interrupts	system call
zombie	parent	new	exec
memory access model	ctrl-c/SIGINT	man pages	foreground process
basic unit of execution	wait	static data segment (SDS)	privileged instructions
disable interrupts	access OS memory	background process	PCB
kernel mode	AMI	interrupt handler	access is restricted
execution state	stack segment	PCB	user mode
life cycle	address space	interrupt vector	main abstraction for protection
PID	process	terminated	ctrl-z/SIGSTP
process state	code segment	external event	loader

Processes: Activity Instructions

- Draw the process life cycle. Label and provide a short description of each state.
- In which state is each of these processes?
 - Process A: waiting for input
 - Process B: waiting for a PID and a PCB
 - Process C: ready to execute
 - Process D: waiting to be reaped
 - Process E: executing a signal handler
 - Process F: waiting for its child
- In what mode is each of those processes?



Directions:

1. Make a copy of the document so that you can edit the drawing.
2. Organize shapes into the process life cycle
3. Add correct labels
4. Place processes in/near their current place in the life cycle

Further Discussion

- What meta-data structure contains information about processes? Name two items found in that data structure.
- Which system call(s) creates a process?
- We learned that operating systems have three main functions: to be a referee, to be an illusionist, and to provide the glue for the system. In no more than thirty words, explain how the process abstraction supports the first two functions.