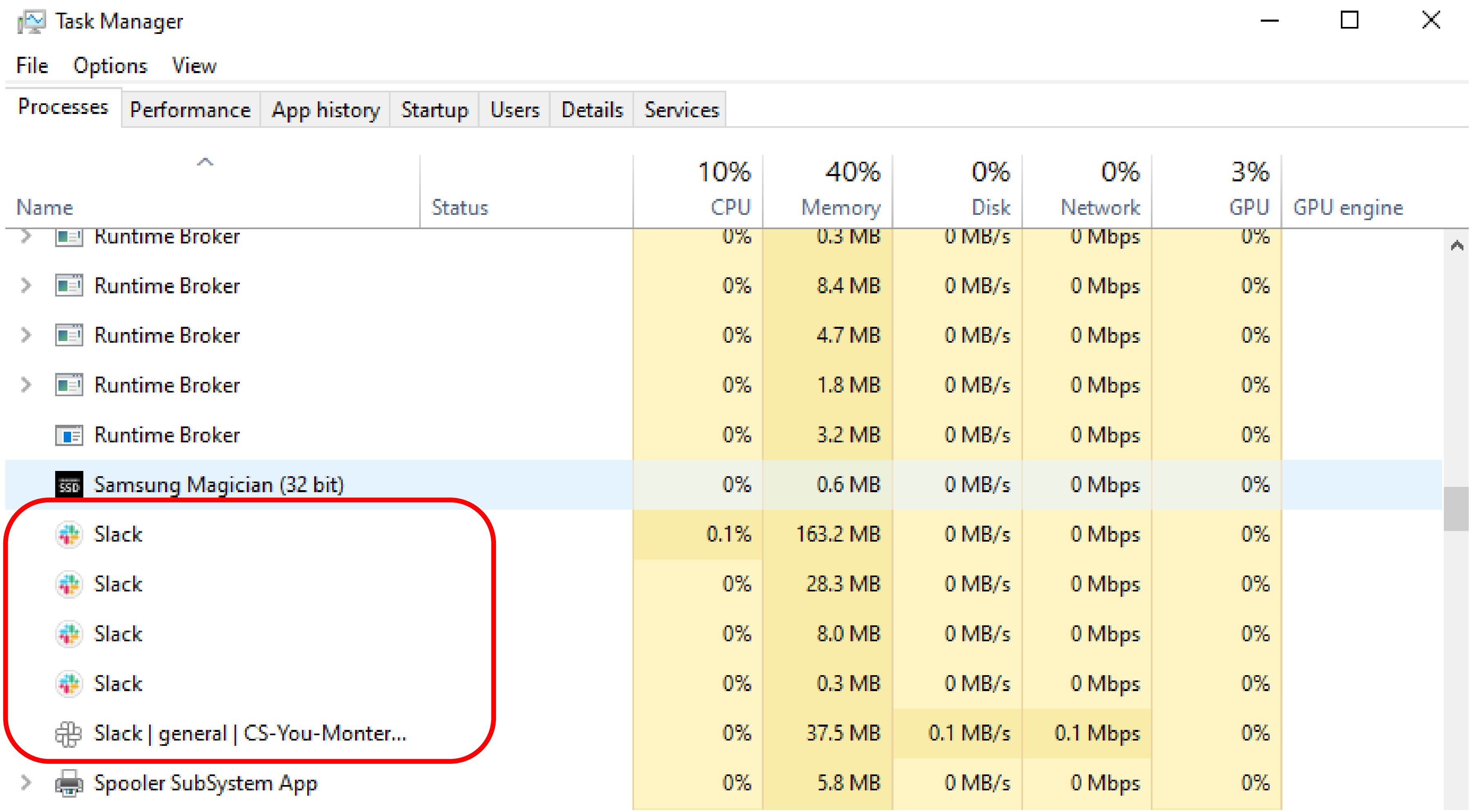
Processes

Glenn Bruns

CSUMB

What's going on here?



Name	Status	10% CPU	40% Memory	0% Disk	0% Network	3% GPU	GPU engine
Runtime Broker		0%	0.3 MB	0 MB/s	0 Mbps	0%	
Runtime Broker		0%	8.4 MB	0 MB/s	0 Mbps	0%	
Runtime Broker		0%	4.7 MB	0 MB/s	0 Mbps	0%	
Runtime Broker		0%	1.8 MB	0 MB/s	0 Mbps	0%	
Runtime Broker		0%	3.2 MB	0 MB/s	0 Mbps	0%	
Samsung Magician (32 bit)		0%	0.6 MB	0 MB/s	0 Mbps	0%	
Slack		0.1%	163.2 MB	0 MB/s	0 Mbps	0%	
Slack		0%	28.3 MB	0 MB/s	0 Mbps	0%	
Slack		0%	8.0 MB	0 MB/s	0 Mbps	0%	
Slack		0%	0.3 MB	0 MB/s	0 Mbps	0%	
Slack | general | CS-You-Monter...		0%	37.5 MB	0.1 MB/s	0.1 Mbps	0%	
Spooler SubSystem App		0%	5.8 MB	0 MB/s	0 Mbps	0%	

Lecture Objectives

At the end of this lecture, you should be able to:

- □ Define ‘process’, and describe the difference between a program and a process
- □ Name some process states
- □ Explain, at a high level, how the OS manages processes
- □ Run bash commands related to processes

Program versus Running Program

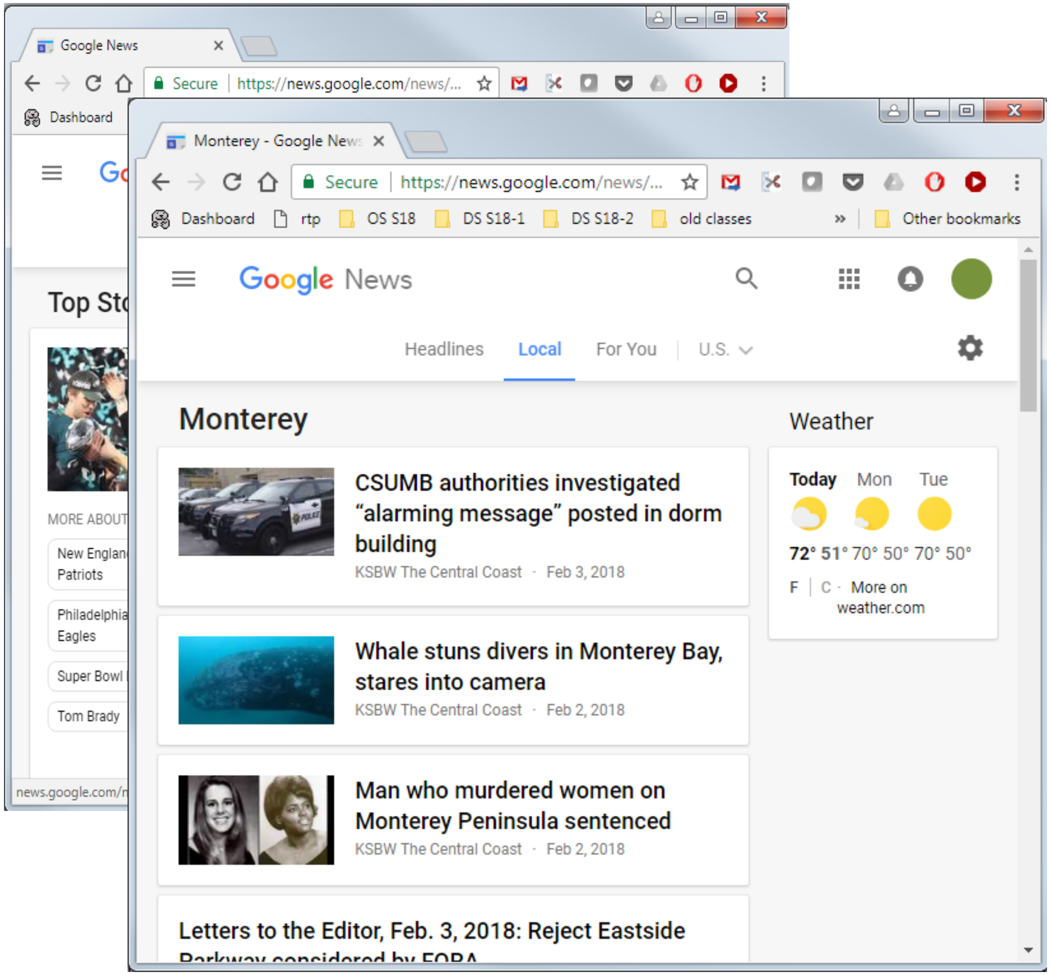
Program

- doesn’t change
- just a bunch of text

Running Program

- has a current “state of execution”:
  - what’s the next statement to run?
  - what are the values in memory?
  - what files are open?
- multiple copies of a program can be running at once

Two browser windows



Google News

Secure | https://news.google.com/news/...

Dashboard

Top Sto

MORE ABOUT

New Englan Patriots

Philadelphia Eagles

Super Bowl

Tom Brady

news.google.com/n

Monterey - Google News

Secure | https://news.google.com/news/...

Dashboard rtp OS S18 DS S18-1 DS S18-2 old classes » Other bookmarks

Google News

Headlines Local For You U.S.

Monterey

CSUMB authorities investigated "alarming message" posted in dorm building
KSBW The Central Coast · Feb 3, 2018

Whale stuns divers in Monterey Bay, stares into camera
KSBW The Central Coast · Feb 2, 2018

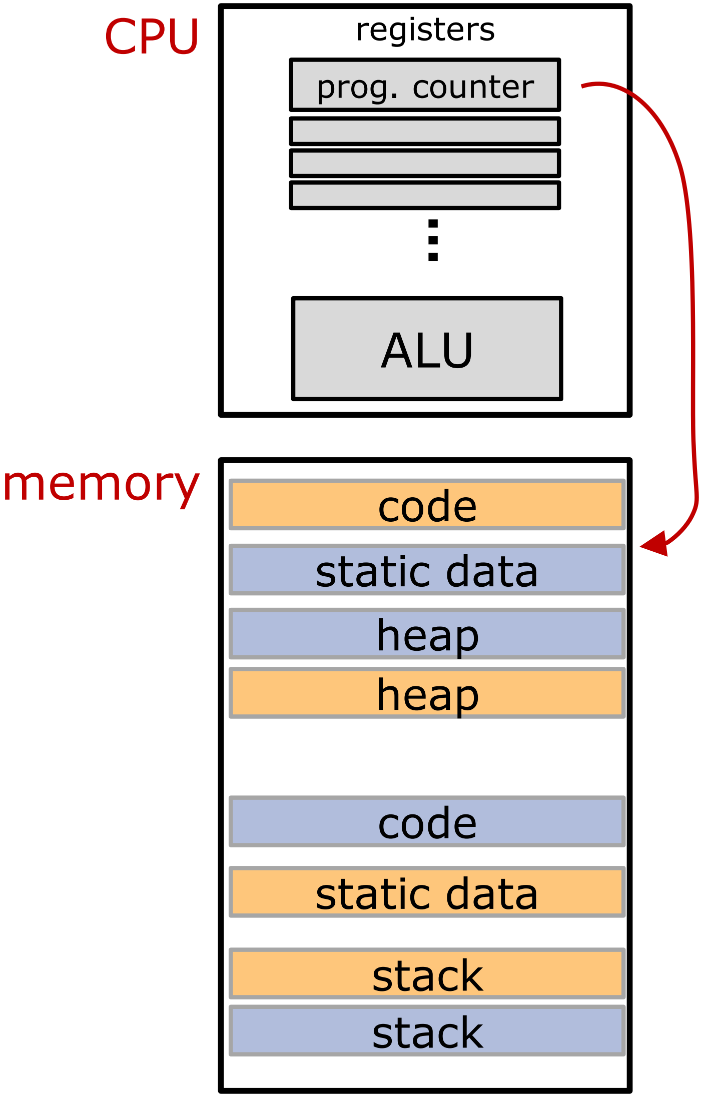
Man who murdered women on Monterey Peninsula sentenced
KSBW The Central Coast · Feb 2, 2018

Letters to the Editor, Feb. 3, 2018: Reject Eastside Parkway considered by FORA

Weather

Today	Mon	Tue
72° 51°	70° 50°	70° 50°

F | C · More on weather.com



What is a process?

A process is a “running program”, or an “executing program”

A process has state, a program doesn't.

An OS manages the processes on a computer.

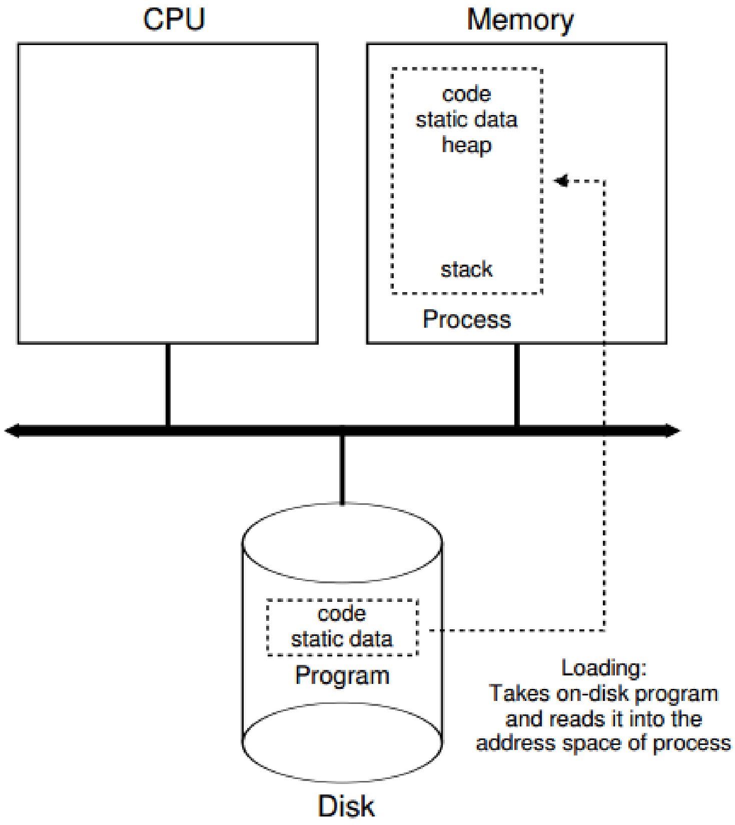
For example:

- create a process
- destroy a process

```
int i1 = 0;
int i2 = 0;
int j = 0;
while (i1 < len1 && i2 < len2) {
  if (a1[i1] < a2[i2]) {
    b[j] = a1[i1];
    i1++;
  } else {
    b[j] = a2[i2];
    i2++;
  }
  j++;
}
```

How does the OS create a process?

1. get the program code off disk, load it into memory
2. load static data into memory
3. allocate memory for stack
4. allocate memory for heap
5. initialize some file descriptors
6. transfer control to the program



Multiprogramming

The OS makes it looks like multiple programs are running at the same time on a single CPU.

This is multiprogramming.

It's another part of the OS's job of process management.

Idea of multiprogramming:

1. run program 1 for a little while
2. pause program 1, run program 2 for a little while
3. pause program 2, resume program 1 and let it run for a little while
4. etc.

Do this really fast and it looks like both programs are running at once

How to pause and resume programs?

Conceptually, to pause:

□ stop process and record its execution state

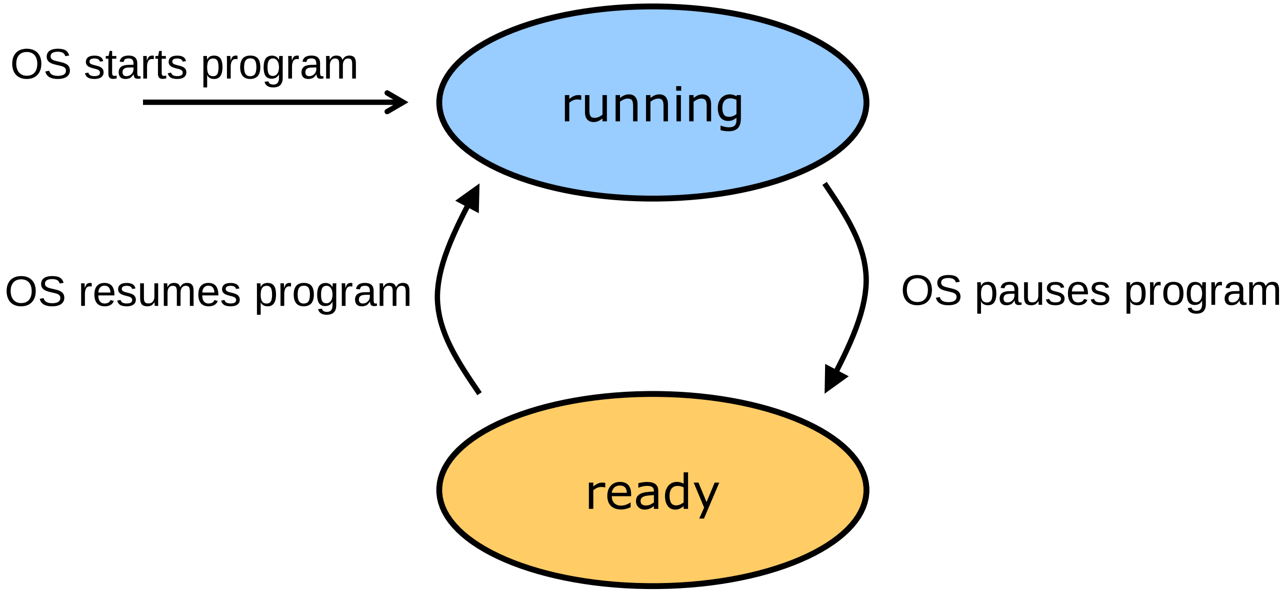
To resume:

□ restore the process execution state, and re-start

We'll get into the details later

Process states

Processes can be running or ready to run, and the OS needs to manage this, too.



We'll see additional process states later.

Multiprogramming

	process 1	process 2
time t = 1	running	ready
t = 2	ready	running
t = 3	running	ready
t = 4	ready	running
t = 5	running	ready

OS data structures for processes mgmt.

For each process there is a data structure containing things like:

- the process id
- the process state
- the process register values
- the size of process memory

In Linux, the data structure is called ‘task_struct’.

When a process is created, a new task_struct is allocated.

Mechanism and Policy

The OS has a mechanism to allow it to stop and start processes.

Once it has a mechanism, it needs a policy to decide how to schedule processes:

- how long should a process be allowed to run?
- which processes should get higher priority?
- how to ensure every process eventually gets a chance to run

Mechanism vs Policy is a recurring theme in operating system design

Listing processes with bash

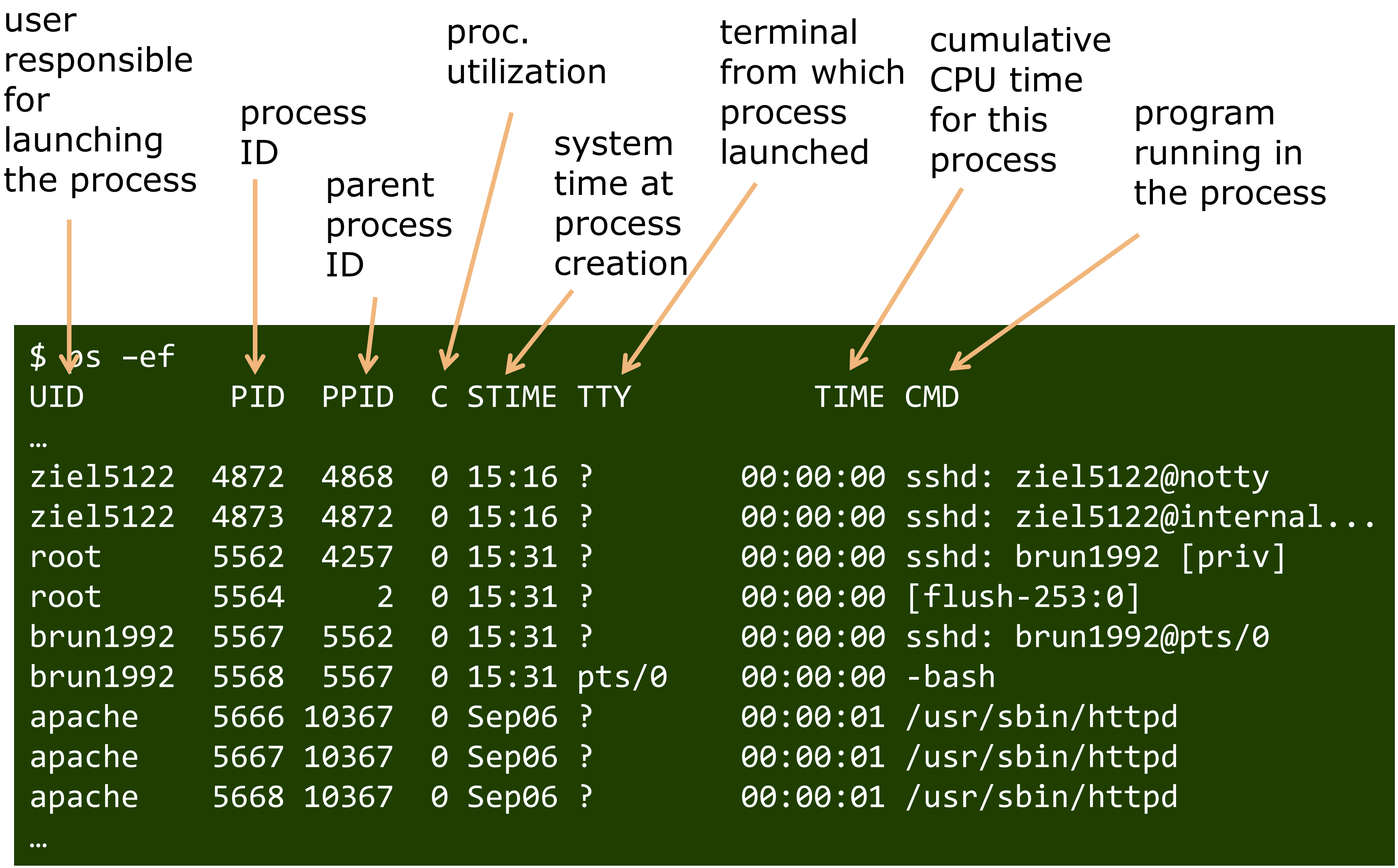
with no options, ps shows processes of current user in current terminal

```
$ ps
  PID TTY          TIME CMD
 5568 pts/0    00:00:00 bash
 5593 pts/0    00:00:00 ps
```

with the –e (all processes) and –f (full output) options:

```
$ ps -ef
…
ziel5122  4872  4868  0 15:16 ?        00:00:00 sshd: ziel5122@notty
ziel5122  4873  4872  0 15:16 ?        00:00:00 sshd: ziel5122@internal...
root      5562  4257  0 15:31 ?        00:00:00 sshd: brun1992 [priv]
root      5564     2  0 15:31 ?        00:00:00 [flush-253:0]
brun1992  5567  5562  0 15:31 ?        00:00:00 sshd: brun1992@pts/0
brun1992  5568  5567  0 15:31 pts/0    00:00:00 -bash
apache    5666 10367  0 Sep06 ?        00:00:01 /usr/sbin/httpd
apache    5667 10367  0 Sep06 ?        00:00:01 /usr/sbin/httpd
apache    5668 10367  0 Sep06 ?        00:00:01 /usr/sbin/httpd
…
```

ps – details



user responsible for launching the process

process ID

parent process ID

proc. utilization

system time at process creation

terminal from which process launched

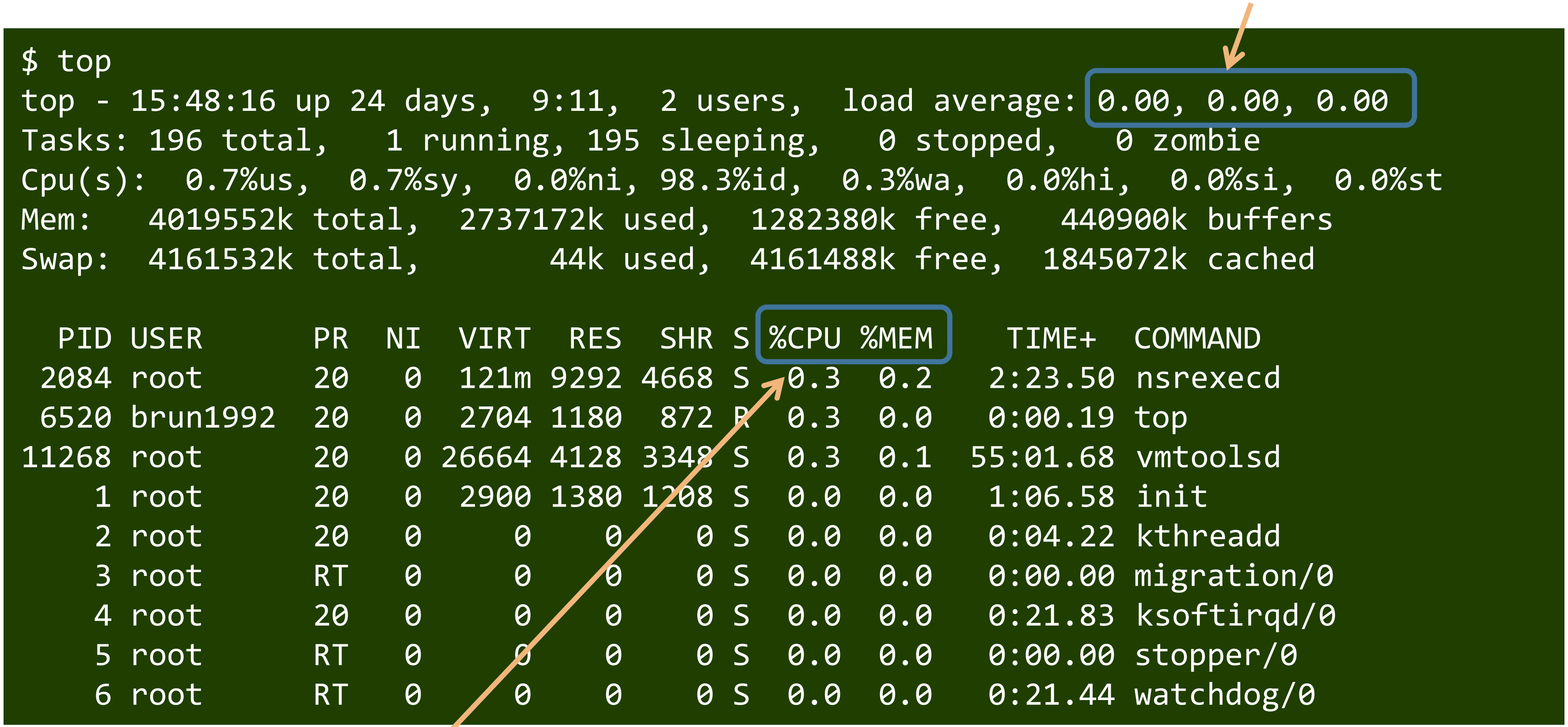
cumulative CPU time for this process

program running in the process

```
$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
…
ziel5122  4872  4868  0 15:16 ?        00:00:00 sshd: ziel5122@notty
ziel5122  4873  4872  0 15:16 ?        00:00:00 sshd: ziel5122@internal...
root      5562  4257  0 15:31 ?        00:00:00 sshd: brun1992 [priv]
root      5564     2  0 15:31 ?        00:00:00 [flush-253:0]
brun1992  5567  5562  0 15:31 ?        00:00:00 sshd: brun1992@pts/0
brun1992  5568  5567  0 15:31 pts/0    00:00:00 -bash
apache    5666 10367  0 Sep06 ?        00:00:01 /usr/sbin/httpd
apache    5667 10367  0 Sep06 ?        00:00:01 /usr/sbin/httpd
apache    5668 10367  0 Sep06 ?        00:00:01 /usr/sbin/httpd
…
```

top – show processes interactively

1, 5, and 15-minute load averages



```
$ top
top - 15:48:16 up 24 days,  9:11,  2 users,  load average: 0.00, 0.00, 0.00
Tasks: 196 total,   1 running, 195 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.7%us,  0.7%sy,  0.0%ni, 98.3%id,  0.3%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4019552k total,  2737172k used,  1282380k free,   440900k buffers
Swap:  4161532k total,       44k used,  4161488k free,  1845072k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 2084 root      20   0  121m 9292 4668 S  0.3  0.2   2:23.50 nsrexecd
 6520 brun1992  20   0  2704 1180  872 R  0.3  0.0   0:00.19 top
11268 root      20   0 26664 4128 3348 S  0.3  0.1  55:01.68 vmtoolsd
    1 root      20   0  2900 1380 1208 S  0.0  0.0   1:06.58 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:04.22 kthreadd
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:21.83 ksoftirqd/0
    5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 stopper/0
    6 root      RT   0     0    0    0 S  0.0  0.0   0:21.44 watchdog/0
```

%CPU: tasks share of the elapsed CPU time since last screen update
%MEM: a task’s currently-used share of available physical memory

kill – ask a process to stop

by default, kill sends a TERM signal (terminate if possible)

```
$ kill 5968
-bash: kill: (5968) - Operation not permitted

$ kill -s HUP 5968
```

or the signal to be sent can be specified

Summary

- □ A process is a running program
- □ A main job of the OS is to manage processes: create them, destroy them, stop them, start them, …
- □ Multi-programming is when the OS runs multiple programs “at once” on a single processor
- □ Mechanism vs Policy
- □ Bash commands for processes