

# Lecture 13: Resource Monitoring Tools

# Putting it all together

- Processes, the Kernel, and Hardware all interact.
- There are three basic types of resource
  - CPU
  - Memory
  - Input/Output (I/O)
- You can also think of the Kernel as a resource that processes have to share
- As an HPC engineer or System Analyst you have to be able to track usage of the system and detect bottlenecks.
- Even so the default settings are usually pretty good

# Overview

- Tracking Processes
- Finding Open Files
- Tracing Program Execution
- Threads
- Resource Monitoring
- Control Groups

```
[matthew@moonshine ~]$ ps | head -n 10
```

PID	TTY	TIME	CMD
160323	pts/1	00:00:00	bash
160357	pts/1	00:00:00	ps
160358	pts/1	00:00:00	head

```
[matthew@moonshine ~]$ ps -A | head -n 10
```

PID	TTY	TIME	CMD
1	?	00:00:22	systemd
2	?	00:00:00	kthreadd
3	?	00:00:00	rcu_gp
4	?	00:00:00	rcu_par_gp
5	?	00:00:00	slub_flushwq
6	?	00:00:00	netns
8	?	00:00:00	kworker/0:0H-
events_highpri			
11	?	00:00:00	mm_percpu_wq
13	?	00:00:00	rcu_tasks_kthre

```
[matthew@moonshine ~]$ ps aux | tail -n 10
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
matthew	168570	0.0	0.0	19044	7176	?	S	07:59	0:00	sshd: matthew@pts/1
matthew	168571	0.0	0.0	7552	4392	pts/1	Ss	07:59	0:00	-bash
root	168576	0.0	0.0	0	0	?	I	07:59	0:00	[kworker/2:2-mm_percpu_wq]
root	168577	0.0	0.0	0	0	?	I	07:59	0:00	[kworker/24:2-events]
root	168609	0.0	0.0	0	0	?	I	07:59	0:00	[kworker/16:0-mm_percpu_wq]
root	168610	0.0	0.0	0	0	?	I	07:59	0:00	[kworker/18:2-mm_percpu_wq]
root	168632	0.0	0.0	15856	8552	?	Ss	08:01	0:00	sshd: [accepted]
root	168648	0.0	0.0	15856	8516	?	Ss	08:03	0:00	sshd: [accepted]
matthew	168651	0.0	0.0	10140	3588	pts/1	R+	08:03	0:00	ps aux
matthew	168652	0.0	0.0	5616	1020	pts/1	S+	08:03	0:00	tail -n 10

`ps aux` is a commonly used command. It shows the user that is running each process. The CPU and Memory usage (RSS is the physical memory usage), and the state of the process (STAT).

The `kworkers` are kernel threads not user space processes.

## PROCESS STATE CODES (from man ps)

D uninterruptible sleep (usually IO)  
I Idle kernel thread  
R running or runnable (on run queue)  
S interruptible sleep (waiting for an event to complete)  
T stopped by job control signal  
t stopped by debugger during the tracing  
W paging (not valid since the 2.6.xx kernel)  
X dead (should never be seen)  
Z defunct ("zombie") process, terminated but not reaped by its parent  
< high-priority (not nice to other users)  
N low-priority (nice to other users)  
L has pages locked into memory (for real-time and custom IO)  
s is a session leader  
l is multi-threaded (using CLONE\_THREAD, like NPTL pthreads do)  
+ is in the foreground process group

**+**: Foreground. A process is in the “foreground” if it is interactive in a shell. I.e. it can send data to the terminal and is listening to the keyboard.

A process can be “backgrounded” with ctrl-z and brought back into the foreground with “fg”.

## Background and Foreground

```
[matthew@moonshine FORTRAN_SMP]$ time OMP_NUM_THREADS=8 ./smp_vecadd 1000000000
^Z
[1]+  Stopped                  OMP_NUM_THREADS=8 ./smp_vecadd 1000000000

real 0m0.872s
user 0m0.000s
sys 0m0.000s
[matthew@moonshine FORTRAN_SMP]$ bg
[1]+ OMP_NUM_THREADS=8 ./smp_vecadd 1000000000 &
[matthew@moonshine FORTRAN_SMP]$ fg
OMP_NUM_THREADS=8 ./smp_vecadd 1000000000
```

Ctrl-z stops a process.

Bg disconnects the process from the terminal

Fg brings it back into the foreground

Notice that “time” gets confused because we suspended the process it was timing.

You can start a process to start in the background with &

```
OMP_NUM_THREADS=8 ./smp_vecadd 1000000000 &
```



## PROCESS STATE CODES

```
$ ps aux | grep smp_vecadd
```

```
Every 2.0s: ps aux | grep
```

```
smp_vecadd
```

```
moonshine: Fri
```

```
Mar 1 08:21:07 2024
```

```
matthew 168863 0.0 0.0 6292 3008 pts/2 S+ 08:20 0:00 watch ps aux | grep smp_vecadd
matthew 169488 113 12.3 11783132 8049524 pts/1 Rl+ 08:25 0:06 ./smp_vecadd 1000000000
matthew 168929 0.0 0.0 6292 1104 pts/2 S+ 08:21 0:00 watch ps aux | grep smp_vecadd
matthew 168930 0.0 0.0 7124 3316 pts/2 S+ 08:21 0:00 sh -c ps aux | grep smp_vecadd
matthew 168932 0.0 0.0 6408 2172 pts/2 S+ 08:21 0:00 grep smp_vecadd
```

```
[matthew@moonshine FORTRAN_SMP]$ time OMP_NUM_THREADS=8 ./smp_vecadd 1000000000
```

```
real 0m7.395s
user 0m10.985s
sys 0m2.613s
```

Rl+: running in the foreground and has multiple threads

## PROCESS STATE CODES

```
$ ps aux | grep smp_vecadd
```

```
Every 2.0s: ps aux | grep
```

```
smp_vecadd
```

```
moonshine: Fri
```

```
Mar 1 08:21:07 2024
```

```
matthew 168863 0.0 0.0 6292 3008 pts/2 S+ 08:20 0:00 watch ps aux | grep smp_vecadd
matthew 169488 113 12.3 11783132 8049524 pts/1 Rl+ 08:25 0:06 ./smp_vecadd 1000000000
matthew 168929 0.0 0.0 6292 1104 pts/2 S+ 08:21 0:00 watch ps aux | grep smp_vecadd
matthew 168930 0.0 0.0 7124 3316 pts/2 S+ 08:21 0:00 sh -c ps aux | grep smp_vecadd
matthew 168932 0.0 0.0 6408 2172 pts/2 S+ 08:21 0:00 grep smp_vecadd
```

```
[matthew@moonshine FORTRAN_SMP]$ time OMP_NUM_THREADS=8 ./smp_vecadd 1000000000
```

```
real 0m7.395s
user 0m10.985s
sys 0m2.613s
```

```
llll
```

```
[matthew@moonshine ~]$ pgrep -l firewalld  
1269 firewalld
```

```
[matthew@moonshine ~]$ ps -A --forest | head -n 20
```

PID	TTY	TIME	CMD
2	?	00:00:00	kthreadd
3	?	00:00:00	\_rcu_gp
4	?	00:00:00	\_rcu_par_gp
5	?	00:00:00	\_slub_flushwq
6	?	00:00:00	\_netns
8	?	00:00:00	\_kworker/0:0H-events_highpri
11	?	00:00:00	\_mm_percpu_wq
13	?	00:00:00	\_rcu_tasks_kthre
14	?	00:00:00	\_rcu_tasks_rude_
15	?	00:00:00	\_rcu_tasks_trace
16	?	00:00:00	\_ksoftirqd/0
17	?	00:00:16	\_pr/tty0
18	?	00:01:30	\_rcu_preempt
19	?	00:00:01	\_migration/0
20	?	00:00:00	\_idle_inject/0
22	?	00:00:00	\_cpuhp/0
23	?	00:00:00	\_cpuhp/1
24	?	00:00:00	\_idle_inject/1
25	?	00:00:01	\_migration/1

# Top command

```
top - 08:43:24 up 7 days, 19:08, 2 users, load average: 0.58, 0.23, 0.09
Tasks: 422 total, 2 running, 420 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.4 us, 0.7 sy, 0.0 ni, 96.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 63774.3 total, 54901.7 free, 7748.5 used, 1844.0 buff/cache
MiB Swap: 32208.0 total, 32208.0 free, 0.0 used. 56025.8 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
138549	matthew	20	0	11.2g	6.1g	2632	R	99.3	9.8	0:05.21	smp_vecadd
138514	matthew	20	0	10844	4248	3384	R	0.7	0.0	0:00.21	top
138276	root	20	0	0	0	0	I	0.3	0.0	0:01.96	kworker/0:2-events
1	root	20	0	174172	18364	10796	S	0.0	0.0	0:18.62	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.11	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
11	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthre
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_
15	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace
16	root	20	0	0	0	0	S	0.0	0.0	0:00.07	ksoftirqd/0
17	root	20	0	0	0	0	S	0.0	0.0	0:14.02	pr/tty0

```

top - 20:07:23 up 9 days, 6:32, 2 users, load average: 0.07, 0.02, 0.00
Tasks: 420 total, 1 running, 419 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 63774.3 total, 61121.4 free, 1505.9 used, 1897.0 buff/cache
MiB Swap: 32208.0 total, 32208.0 free, 0.0 used. 62268.4 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
159551	root	20	0	0	0	0	I	0.3	0.0	0:08.56	kworker+
<b>160445</b>	<b>matthew</b>	<b>20</b>	<b>0</b>	<b>10844</b>	<b>4248</b>	<b>3380</b>	<b>R</b>	<b>0.3</b>	<b>0.0</b>	<b>0:00.06</b>	<b>top</b>
160447	root	20	0	18896	10720	9136	S	0.3	0.0	0:00.01	sshd
1	root	20	0	174172	19904	10800	S	0.0	0.0	0:22.73	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.13	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par+
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_fl+
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker+
11	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_perc+
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tas+
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tas+
15	root	0	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tas+
16	root	0	0	0	0	0	I	0.0	0.0	0:00.09	ksoftir+
17	root	0	0	0	0	0	I	0.0	0.0	0:16.60	pr/tty0
18	root	20	0	0	0	0	I	0.0	0.0	1:30.82	rcu_pre+

Sort by CPU (default)

top - 20:08:03 up 9 days, 6:33, 2 users, load average: 0.04, 0.01, 0.00 **M**  
 Tasks: 419 total, 1 running, 418 sleeping, 0 stopped, 0 zombie  
 %Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
 MiB Mem : 63774.3 total, 61120.9 free, 1506.4 used, 1897.1 buff/cache  
 MiB Swap: 32208.0 total, 32208.0 free, 0.0 used. 62268.0 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
954	root	20	0	149308	121304	119872	S	0.0	0.2	1:19.82	systemd+
1269	root	20	0	127964	43420	17724	S	0.0	0.1	0:00.40	firewal+
1395	root	20	0	319744	25148	23392	S	0.0	0.0	0:55.71	rsyslogd
1364	polkitd	20	0	2917988	23912	18548	S	0.0	0.0	0:00.16	polkitd
1280	root	20	0	257360	23724	16644	S	0.0	0.0	2:55.26	Network+
1	root	20	0	174172	19904	10800	S	0.0	0.0	0:22.73	systemd
116604	matthew	20	0	22496	13916	10628	S	0.0	0.0	0:00.14	systemd
971	root	20	0	35052	13240	9104	S	0.0	0.0	0:00.83	systemd+
1272	root	20	0	28788	13192	8992	S	0.0	0.0	0:00.78	systemd+
158798	root	20	0	18996	11664	9768	S	0.0	0.0	0:00.18	sshd
160314	root	20	0	18856	11600	9712	S	0.0	0.0	0:00.17	sshd
1330	root	20	0	15852	9304	8000	S	0.0	0.0	0:21.69	sshd
160449	root	20	0	15856	9292	8000	S	0.0	0.0	0:00.00	sshd
1273	root								0.0	0:09.79	systemd+
116606	matthew								0.0	0:00.00	(sd-pam)
160322	matthew								0.0	0:00.04	sshd
158802	matthew	20	0	18908	7236	5340	S	0.0	0.0	0:00.05	sshd

Sort by memory usage



```
top - 20:09:45 up 9 days, 6:35, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 418 total, 1 running, 417 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 63774.3 total, 61121.5 free, 1505.7 used, 1897.1 buff/cache
MiB Swap: 32208.0 total, 32208.0 free, 0.0 used. 62268.6 avail Mem
```

u

```
Which user (blank for all) matthew
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
116604	matthew	20	0	22496	13916	10628	S	0.0	0.0	0:00.14	systemd
116606	matthew	20	0	174392	8236	0	S	0.0	0.0	0:00.00	(sd-pam)
160322	matthew	20	0	19040	7336	5440	S	0.0	0.0	0:00.05	sshd
158802	matthew	20	0	18908	7236	5340	S	0.0	0.0	0:00.05	sshd
120028	matthew	20	0	81516	5152	2856	S	0.0	0.0	0:00.00	scdaemon
158805	matthew	20	0	7552	4400	3616	S	0.0	0.0	0:00.02	bash
160323	matthew	20	0	7652	4372	3592	S	0.0	0.0	0:00.02	bash
<b>160445</b>	<b>matthew</b>	<b>20</b>	<b>0</b>	<b>10844</b>	<b>4248</b>	<b>3380</b>	<b>R</b>	<b>0.3</b>	<b>0.0</b>	<b>0:00.41</b>	<b>top</b>
120026	matthew	20	0	228664	2608	2248	S	0.0	0.0	0:02.73	gpg-age+

Filter by user



# Fields Management for window 1:Def, whose current sort field is %CPU

Navigate with Up/Dn, Right selects for move then <Enter> or Left commits, 'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!

* PID	= Process Id	PGRP	= Process Group	OOMs	= OOMEM Score c
* USER	= Effective Use	TTY	= Controlling T	ENVIRON	= Environment v
* PR	= Priority	TPGID	= Tty Process G	vMj	= Major Faults
* NI	= Nice Value	SID	= Session Id	vMn	= Minor Faults
* VIRT	= Virtual Image	nTH	= Number of Thr	USED	= Res+Swap Size
* RES	= Resident Size	P	= Last Used Cpu	nsIPC	= IPC namespace
* SHR	= Shared Memory	TIME	= CPU Time	nsMNT	= MNT namespace
* S	= Process Statu	SWAP	= Swapped Size	nsNET	= NET namespace
* %CPU	= CPU Usage	CODE	= Code Size (Ki	nsPID	= PID namespace
* %MEM	= Memory Usage	DATA	= Data+Stack (K	nsUSER	= USER namespac
* TIME+	= CPU Time, hun	nMaj	= Major Page Fa	nsUTS	= UTS namespace
* COMMAND	= Command Name/	nMin	= Minor Page Fa	LXC	= LXC container
PPID	= Parent Proces	nDRT	= Dirty Pages C	RSan	= RES Anonymous
UID	= Effective Use	WCHAN	= Sleeping in F	RSfd	= RES File-base
RUID	= Real User Id	Flags	= Task Flags <s	RSlk	= RES Locked (K
RUSER	= Real User Nam	CGROUPS	= Control Group	RSsh	= RES Shared (K
SUID	= Saved			IAME	= Control Group
SUSER	= Saved				= Last Used NUM
GID	= Group				
GROUP	= Group Name	OOMa	= OOMEM Adjustm		

List all the filter options

```
[matthew@moonshine ~]$ sudo yum install lsof
```

```
Dependencies resolved.
```

```
=====
```

Package	Architecture	Version	Repository	Size
Installing:				
lsof	x86_64	4.94.0-3.el9	baseos	238 k

```
=====
```

```
Total                               238 kB/s | 238 kB      00:00
```

```
Running transaction check
```

```
Transaction check succeeded.
```

```
Running transaction test
```

```
Transaction test succeeded.
```

```
Running transaction
```

```
  Preparing      :                               1/1  
  Installing     : lsof-4.94.0-3.el9.x86_64     1/1  
  Running scriptlet: lsof-4.94.0-3.el9.x86_64   1/1  
  Verifying      : lsof-4.94.0-3.el9.x86_64     1/1
```

```
Installed:
```

```
  lsof-4.94.0-3.el9.x86_64
```

```
Complete!
```

```
[matthew@moonshine ~]$
```

# Finding open files

```
moonshine ~]$ sudo lsof | head -n 10
```

```
COMMAND      PID    USER   FD     TYPE  DEVICE  SIZE/OFF      NODE NAME
systemd       1     root   cwd     DIR   253,0    235        128 /
systemd       1     root   rtd     DIR   253,0    235        128 /
systemd       1     root   txt     REG   253,0   102128   134486352 /usr/lib/systemd/systemd
systemd       1     root   mem     REG   253,0   637880   201328860 /usr/lib64/libpcre2-8.so.0.11.0
systemd       1     root   mem     REG   253,0   904680   201328591 /usr/lib64/libm.so.6
systemd       1     root   mem     REG   253,0   882384   201328723 /usr/lib64/libzstd.so.1.5.1
systemd       1     root   mem     REG   253,0  4482528   202761030 /usr/lib64/libcrypto.so.3.0.7
systemd       1     root   mem     REG   253,0   581551    67415145/etc/selinux/targeted/contexts/files/file_contexts.bin
systemd       1     root   mem     REG   253,0  1293840   201328831 /usr/lib64/libp11-kit.so.0.3.0
[matthew@moonshine ~]$
```

## Finding open files

(And remember everything in Linux is a file)

```
moonshine ~]$ sudo lsof | head -n 10
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
<b>systemd</b>	1	root	cwd	DIR	253,0	235	128 /	
systemd	1	root	rtd	DIR	253,0	235	128 /	
systemd	1	root	txt	REG	253,0	102128	134486352	/usr/lib/systemd/systemd
systemd	1	root	mem	REG	253,0	637880	201328860	/usr/lib64/libpcre2-8.so.0.11.0
systemd	1	root	mem	REG	253,0	904680	201328591	/usr/lib64/libm.so.6
systemd	1	root	mem	REG	253,0	882384	201328723	/usr/lib64/libzstd.so.1.5.1
systemd	1	root	mem	REG	253,0	4482528	202761030	/usr/lib64/libcrypto.so.3.0.7
systemd	1	root	mem	REG	253,0	581551	67415145	/etc/selinux/targeted/contexts/files/file_contexts.bin
systemd	1	root	mem	REG	253,0	1293840	201328831	/usr/lib64/libp11-kit.so.0.3.0

```
[matthew@moonshine ~]$
```

## Finding open files

The name of the process that owns the file.

```
moonshine ~]$ sudo lsof | head -n 10
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
systemd	1	root	cwd	DIR	253,0	235	128	/
systemd	1	root	rtd	DIR	253,0	235	128	/
systemd	1	root	txt	REG	253,0	102128	134486352	/usr/lib/systemd/systemd
systemd	1	root	mem	REG	253,0	637880	201328860	/usr/lib64/libpcre2-8.so.0.11.0
systemd	1	root	mem	REG	253,0	904680	201328591	/usr/lib64/libm.so.6
systemd	1	root	mem	REG	253,0	882384	201328723	/usr/lib64/libzstd.so.1.5.1
systemd	1	root	mem	REG	253,0	4482528	202761030	/usr/lib64/libcrypto.so.3.0.7
systemd	1	root	mem	REG	253,0	581551	67415145	/etc/selinux/targeted/contexts/files/file_contexts.bin
systemd	1	root	mem	REG	253,0	1293840	201328831	/usr/lib64/libp11-kit.so.0.3.0

```
[matthew@moonshine ~]$
```

# Finding open files

## (The Process ID (PID) that owns the file)

```
moonshine ~]$ sudo lsof | head -n 10
```

```
COMMAND      PID    USER   FD          TYPE    DEVICE  SIZE/OFF      NODE NAME
systemd       1     root   cwd          DIR     253,0    235        128 /
systemd       1     root   rtd          DIR     253,0    235        128 /
systemd       1     root   txt          REG     253,0   102128   134486352 /usr/lib/systemd/systemd
systemd       1     root   mem          REG     253,0   637880   201328860 /usr/lib64/libpcre2-8.so.0.11.0
systemd       1     root   mem          REG     253,0   904680   201328591 /usr/lib64/libm.so.6
systemd       1     root   mem          REG     253,0   882384   201328723 /usr/lib64/libzstd.so.1.5.1
systemd       1     root   mem          REG     253,0  4482528   202761030 /usr/lib64/libcrypto.so.3.0.7
systemd       1     root   mem          REG     253,0   581551   67415145/etc/selinux/targeted/contexts/files/file_contexts.bin
systemd       1     root   mem          REG     253,0  1293840   201328831 /usr/lib64/libp11-kit.so.0.3.0
[matthew@moonshine ~]$
```

## Finding open files

(The user running the process that owns the file)



```
moonshine ~]$ sudo lsof | head -n 10
```

FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
<b>cwd</b>	DIR	253,0	235	128 /	
rtd	DIR	253,0	235	128 /	
txt	REG	253,0	102128	134486352	/usr/lib/systemd/systemd
mem	REG	253,0	637880	201328860	/usr/lib64/libpcre2-8.so.0.11.0
mem	REG	253,0	904680	201328591	/usr/lib64/libm.so.6
mem	REG	253,0	882384	201328723	/usr/lib64/libzstd.so.1.5.1
mem	REG	253,0	4482528	202761030	/usr/lib64/libcrypto.so.3.0.7
mem	REG	253,0	581551	67415145	/etc/selinux/targeted/contexts/files/file_contexts.bin
mem	REG	253,0	1293840	201328831	/usr/lib64/libp11-kit.so.0.3.0

**cwd – Current Working Directory**

**Finding open files  
(File Descriptor (FD))**

```
moonshine ~]$ sudo lsof | head -n 10
```

FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
cwd	DIR	253,0	235	128 /	
rtd	DIR	253,0	235	128 /	
txt	REG	253,0	102128	134486352	/usr/lib/systemd/systemd
mem	REG	253,0	637880	201328860	/usr/lib64/libpcre2-8.so.0.11.0
mem	REG	253,0	904680	201328591	/usr/lib64/libm.so.6
mem	REG	253,0	882384	201328723	/usr/lib64/libzstd.so.1.5.1
mem	REG	253,0	4482528	202761030	/usr/lib64/libcrypto.so.3.0.7
mem	REG	253,0	581551	67415145	/etc/selinux/targeted/contexts/files/file_contexts.bin
mem	REG	253,0	1293840	201328831	/usr/lib64/libp11-kit.so.0.3.0

rtd – root directory

# Finding open files (File Descriptor (FD))



```
moonshine ~]$ sudo lsof | head -n 10
```

FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
cwd	DIR	253,0	235	128 /	
rtd	DIR	253,0	235	128 /	
txt	REG	253,0	102128	134486352	/usr/lib/systemd/systemd
mem	REG	253,0	637880	201328860	/usr/lib64/libpcre2-8.so.0.11.0
mem	REG	253,0	904680	201328591	/usr/lib64/libm.so.6
mem	REG	253,0	882384	201328723	/usr/lib64/libzstd.so.1.5.1
mem	REG	253,0	4482528	202761030	/usr/lib64/libcrypto.so.3.0.7
mem	REG	253,0	581551	67415145	/etc/selinux/targeted/contexts/files/file_contexts.bin
mem	REG	253,0	1293840	201328831	/usr/lib64/libp11-kit.so.0.3.0

txt – program instructions (remember the txt section of ELF)

## Finding open files (File Descriptor (FD))

```
moonshine ~]$ sudo lsof | head -n 10
```

FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
cwd	DIR	253,0	235	128 /	
rtd	DIR	253,0	235	128 /	
txt	REG	253,0	102128	134486352	/usr/lib/systemd/systemd
mem	REG	253,0	637880	201328860	/usr/lib64/libpcre2-8.so.0.11.0
mem	REG	253,0	904680	201328591	/usr/lib64/libm.so.6
mem	REG	253,0	882384	201328723	/usr/lib64/libzstd.so.1.5.1
mem	REG	253,0	4482528	202761030	/usr/lib64/libcrypto.so.3.0.7
mem	REG	253,0	581551	67415145	/etc/selinux/targeted/contexts/files/file_contexts.bin
mem	REG	253,0	1293840	201328831	/usr/lib64/libp11-kit.so.0.3.0

mem – file mapped to RAM. A library in memory for example.

## Finding open files (File Descriptor (FD))

```
moonshine ~]$ sudo lsof | head -n 10
```

FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
cwd	DIR	253,0	235	128 /	
rtd	DIR	253,0	235	128 /	
txt	REG	253,0	102128	134486352	/usr/lib/systemd/systemd
mem	REG	253,0	637880	201328860	/usr/lib64/libpcre2-8.so.0.11.0
mem	REG	253,0	904680	201328591	/usr/lib64/libm.so.6
mem	REG	253,0	882384	201328723	/usr/lib64/libzstd.so.1.5.1
mem	REG	253,0	4482528	202761030	/usr/lib64/libcrypto.so.3.0.7
mem	REG	253,0	581551	67415145	/etc/selinux/targeted/contexts/files/file_contexts.bin
mem	REG	253,0	1293840	201328831	/usr/lib64/libp11-kit.so.0.3.0

Node – The inode number of the file

# Finding open files (File Descriptor (FD))

```
[matthew@moonshine ~]$ lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda                                  8:0     0   931G  0 disk
├─sda1                               8:1     0   600M  0 part /boot/efi
├─sda2                               8:2     0    1G   0 part /boot
└─sda3                               8:3     0 929.4G  0 part
   ├─rl_dhcp52-root                 253:0     0    70G   0 lvm  /
   ├─rl_dhcp52-swap                 253:1     0  31.5G  0 lvm  [SWAP]
   └─rl_dhcp52-home                 253:2     0   828G  0 lvm  /home
sdb                                  8:16     1  14.5G  0 disk
├─sdb1                              8:17     1    1.6G  0 part
└─sdb2                              8:18     1    6.9M  0 part
sr0                                  11:0     1  1024M  0 rom
```

Remember we can use the `lsblk` command to get the drive IDs.

## Finding open files (File Descriptor (FD))

```
[matthew@moonshine ~]$ sudo lsof /run | head -n 10
COMMAND      PID    USER   FD   TYPE    DEVICE  SIZE/OFF  NODE  NAME
systemd      1     root   123u  FIFO    0,24      0t0    650  /run/initctl
systemd      1     root   133u  FIFO    0,24      0t0    645  /run/dmeventd-server
systemd      1     root   134u  FIFO    0,24      0t0    646  /run/dmeventd-client
systemd-j    954    root   mem    REG     0,24  47201216  2371
/run/log/journal/3947007ae3f04ecb9b51072a0c1abde1/system@a5566822de9b428fac68a14b2212ca10-000000000002d576-000612617656d443.journal
systemd-j    954    root   mem    REG     0,24  25165824  2523
/run/log/journal/3947007ae3f04ecb9b51072a0c1abde1/system.journal
systemd-j    954    root   mem    REG     0,24      8    59  /run/systemd/journal/kernel-seqnum
systemd-j    954    root   16u    REG     0,24  25165824  2523
/run/log/journal/3947007ae3f04ecb9b51072a0c1abde1/system.journal
systemd-j    954    root   25u    REG     0,24  47201216  2371
/run/log/journal/3947007ae3f04ecb9b51072a0c1abde1/system@a5566822de9b428fac68a14b2212ca10-000000000002d576-000612617656d443.journal
rpcbind     1243    rpc     8rW   REG     0,24      0  1597 /run/rpcbind/rpcbind.lock
```

We can see what processes have files open in a particular directory.

Let's see what sshd is up to. First get the PIDs associated with sshd.

```
[matthew@moonshine ~]$ pgrep sshd
```

```
1330
```

```
158798
```

```
158802
```

```
160314
```

```
160322
```

```
162187
```

```
162188
```

Let's see what sshd is up to. First get the PIDs associated with sshd.

```
[matthew@moonshine ~]$ sudo lsof -p 1330 | tail -n 10
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
sshd	1330	root	mem	REG	253,0	915816	203042668	/usr/lib64/libsystemd.so.0.35.0
sshd	1330	root	mem	REG	253,0	69472	201331800	/usr/lib64/libpam.so.0.85.1
sshd	1330	root	mem	REG	253,0	134632	201328799	/usr/lib64/libaudit.so.1.0.0
sshd	1330	root	mem	REG	253,0	845312	201328584	/usr/lib64/ld-linux-x86-64.so.2
sshd	1330	root	0r	CHR	1,3	0t0	4	/dev/null
sshd	1330	root	1u	unix	0xffff9e68c6256e80	0t0	51201	type=STREAM (CONNECTED)
sshd	1330	root	2u	unix	0xffff9e68c6256e80	0t0	51201	type=STREAM (CONNECTED)
sshd	1330	root	3u	IPv4	51216	0t0	TCP	*:ssh (LISTEN)
sshd	1330	root	4u	IPv6	51218	0t0	TCP	*:ssh (LISTEN)
sshd	1330	root	6r	FIFO	0,13	0t0	3124918	pipe

The “files” the ssh daemon has open include the ssh network sockets that the daemon uses to communicate.

The types are IPv4 and IPv6 TCP sockets. “u” is read/write and r is read only.

# Current Assignments

- **Homework 3 is due 9:00am Tuesday Mar 5th (tomorrow).**
- Make sure you take advantage of our office hours!

20 /22 students have started the homework.



# Network troubleshooting

```
[matthew@moonshine ~]$ sudo systemctl start echod.socket
```

```
[matthew@moonshine ~]$ systemctl status echod.socket
```

- echod.socket - Echo server

Loaded: loaded (/usr/lib/systemd/system/echod.socket; disabled; preset: disabled)

Active: active (listening) since Thu 2024-02-29 22:33:38 CST; 4s ago

Until: Thu 2024-02-29 22:33:38 CST; 4s ago

Listen: [::]:4444 (Stream)

Accepted: 0; Connected: 0;

Tasks: 0 (limit: 407899)

Memory: 8.0K

CPU: 818us

CGroup: /system.slice/echod.socket

# Network troubleshooting

```
[matthew@moonshine ~]$ sudo lsof -i TCP:22
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
sshd	1330	root	3u	IPv4	51216	0t0	TCP	*:ssh (LISTEN)
sshd	1330	root	4u	IPv6	51218	0t0	TCP	*:ssh (LISTEN)
sshd	158798	root	4u	IPv4	3051569	0t0	TCP	moonshine:ssh->dhcp68.carc.unm.edu:32565 (ESTABLISHED)
sshd	158802	matthew	4u	IPv4	3051569	0t0	TCP	moonshine:ssh->dhcp68.carc.unm.edu:32565 (ESTABLISHED)
sshd	160314	root	4u	IPv4	3086435	0t0	TCP	moonshine:ssh->fricke.co.uk:53385 (ESTABLISHED)
sshd	160322	matthew	4u	IPv4	3086435	0t0	TCP	moonshine:ssh->fricke.co.uk:53385 (ESTABLISHED)

Check to see what “files” the SSH daemon are open to see who is connected and where they connected from.

# Network troubleshooting

```
[matthew@moonshine ~]$ sudo systemctl start echod.socket
```

```
[matthew@moonshine ~]$ sudo lsof -i TCP:4444
```

```
COMMAND PID USER   FD   TYPE    DEVICE  SIZE/OFF  NODE NAME
systemd   1 root   40u  IPv6  3123239      0t0  TCP *:krb524 (LISTEN)
```

```
[matthew@moonshine ~]$ socat - TCP:localhost:4444
```

In another terminal open a connection to port 4444. Leave it open.

# Network troubleshooting

```
[matthew@moonshine ~]$ sudo lsof -i TCP:4444
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
systemd	1	root	20u	IPv6	3123258	0t0	TCP	localhost:krb524->localhost:46892 (ESTABLISHED)
systemd	1	root	40u	IPv6	3123239	0t0	TCP	*:krb524 (LISTEN)
socat	162382	matthew	5u	IPv4	3125403	0t0	TCP	localhost:46892->localhost:krb524 (ESTABLISHED)
echo.py	162384	root	0u	IPv6	3123258	0t0	TCP	localhost:krb524->localhost:46892 (ESTABLISHED)
echo.py	162384	root	1u	IPv6	3123258	0t0	TCP	localhost:krb524->localhost:46892 (ESTABLISHED)
echo.py	162384	root	2u	IPv6	3123258	0t0	TCP	localhost:krb524->localhost:46892 (ESTABLISHED)

```
mfricke@hopper:~ $ module load socat  
mfricke@hopper:~ $ socat - TCP:129.24.245.16:4444
```

In another terminal login to hopper and open a connection to port 4444 on your server. Leave it open.

# Network troubleshooting

```
[matthew@moonshine ~]$ sudo lsof -i TCP:4444
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
systemd	1	root	20u	IPv6	3090862	0t0	TCP	moonshine:krb524->hopper.alliance.unm.edu:47840 (ESTABLISHED)
systemd	1	root	40u	IPv6	3123239	0t0	TCP	*:krb524 (LISTEN)
echo.py	162481	root	0u	IPv6	3090862	0t0	TCP	moonshine:krb524->hopper.alliance.unm.edu:47840 (ESTABLISHED)
echo.py	162481	root	1u	IPv6	3090862	0t0	TCP	moonshine:krb524->hopper.alliance.unm.edu:47840 (ESTABLISHED)
echo.py	162481	root	2u	IPv6	3090862	0t0	TCP	moonshine:krb524->hopper.alliance.unm.edu:47840 (ESTABLISHED)

# Network troubleshooting – show all TCP connections

```
[matthew@moonshine ~]$ sudo lsof -i TCP
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
systemd	1	root	20u	IPv6	3090866	0t0	TCP	moonshine:krb524->hopper.alliance.unm.edu:54190 (ESTABLISHED)
systemd	1	root	40u	IPv6	3123239	0t0	TCP	*:krb524 (LISTEN)
systemd	1	root	102u	IPv4	37973	0t0	TCP	*:sunrpc (LISTEN)
systemd	1	root	104u	IPv6	40991	0t0	TCP	*:sunrpc (LISTEN)
rpcbind	1243	rpc	4u	IPv4	37973	0t0	TCP	*:sunrpc (LISTEN)
rpcbind	1243	rpc	6u	IPv6	40991	0t0	TCP	*:sunrpc (LISTEN)
sshd	1330	root	3u	IPv4	51216	0t0	TCP	*:ssh (LISTEN)
sshd	1330	root	4u	IPv6	51218	0t0	TCP	*:ssh (LISTEN)
sshd	158798	root	4u	IPv4	3051569	0t0	TCP	moonshine:ssh->dhcp68.carc.unm.edu:32565 (ESTABLISHED)
sshd	158802	matthew	4u	IPv4	3051569	0t0	TCP	moonshine:ssh->dhcp68.carc.unm.edu:32565 (ESTABLISHED)
sshd	160314	root	4u	IPv4	3086435	0t0	TCP	moonshine:ssh->fricke.co.uk:53385 (ESTABLISHED)
sshd	160322	matthew	4u	IPv4	3086435	0t0	TCP	moonshine:ssh->fricke.co.uk:53385 (ESTABLISHED)
sshd	162339	root	4u	IPv4	3125139	0t0	TCP	moonshine:ssh->fricke.co.uk:55112 (ESTABLISHED)
sshd	162343	matthew	4u	IPv4	3125139	0t0	TCP	moonshine:ssh->fricke.co.uk:55112 (ESTABLISHED)



# Network troubleshooting – show all UDP connections

```
[matthew@moonshine ~]$ sudo lsof -i UDP
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
systemd	1	root	103u	IPv4	40985	0t0	UDP	*:sunrpc
systemd	1	root	105u	IPv6	36890	0t0	UDP	*:sunrpc
rpcbind	1243	rpc	5u	IPv4	40985	0t0	UDP	*:sunrpc
rpcbind	1243	rpc	7u	IPv6	36890	0t0	UDP	*:sunrpc
chronyd	1277	chrony	5u	IPv4	21660	0t0	UDP	localhost:323
chronyd	1277	chrony	6u	IPv6	21661	0t0	UDP	localhost:323

## Check which files a user has open:

```
[matthew@moonshine ~]$ sudo lsof -u matthew | tail -n 10
tail      162571 matthew  mem      REG      253,0      77    134368506 /usr/lib/locale/en_US.utf8/LC_NAME
tail      162571 matthew  mem      REG      253,0     167    201328552 /usr/lib/locale/en_US.utf8/LC_ADDRESS
tail      162571 matthew  mem      REG      253,0      59    201328557 /usr/lib/locale/en_US.utf8/LC_TELEPHONE
tail      162571 matthew  mem      REG      253,0      23    201328555 /usr/lib/locale/en_US.utf8/LC_MEASUREMENT
tail      162571 matthew  mem      REG      253,0    26988    67120394 /usr/lib64/gconv/gconv-modules.cache
tail      162571 matthew  mem      REG      253,0   845312   201328584 /usr/lib64/ld-linux-x86-64.so.2
tail      162571 matthew  mem      REG      253,0      369    201328554 /usr/lib/locale/en_US.utf8/LC_IDENTIFICATION
tail      162571 matthew   0r      FIFO      0,13      0t0     3101872 pipe
tail      162571 matthew   1u      CHR      136,1      0t0         4 /dev/pts/1
tail      162571 matthew   2u      CHR      136,1      0t0         4 /dev/pts/1
```

Notice the pseudoterminal character (CHR) read/write connections.

# Tracing Execution

# Execution tracing

```
[matthew@moonshine ~]$ sudo yum install strace
Last metadata expiration check: 2:44:44 ago on Thu 29 Feb 2024 08:23:04 PM CST.
Dependencies resolved.
```

```
=====
Package                Architecture          Version                R
=====
Installing:
strace                  x86_64                5.18-2.el9            b
```

## Transaction Summary

```
=====
Install 1 Package
```

```
Total download size: 1.4 M
Installed size: 2.9 M
Is this ok [y/N]:
```

# Install strace

# Execution tracing

Install strace – strace prints all the system calls a program makes

```
[matthew@moonshine ~]$ strace ls
execve("/usr/bin/ls", ["ls"], 0x7ffe857e1a70 /* 31 vars */) = 0
brk(NULL) = 0x55b68aefa000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc59a23cb0) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
opentat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=31379, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 31379, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fa980ad9000
close(3) = 0
opentat(AT_FDCWD, "/lib64/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0Pp\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=175552, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fa980ad7000
mmap(NULL, 181896, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fa980aaa000
mmap(0x7fa980ab0000, 110592, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7fa980ab0000
mmap(0x7fa980acb000, 32768, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x21000) = 0x7fa980acb000
mmap(0x7fa980ad3000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7fa980ad3000
mmap(0x7fa980ad5000, 5768, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fa980ad5000
close(3) = 0
write(1, " echo.py\t\t\t gnu_compiler_optimi"..., 103 echo.py gnu_compiler_optimisations OpenMP-offload-train
) = 103
close(1) = 0
close(2) = 0
exit_group(0) = ?
+++ exited with 0 +++
```



# Execution tracing

Install strace – strace prints all the system calls a program makes

```
[matthew@moonshine ~]$ strace ls
execve("/usr/bin/ls", ["ls"], 0x7ffe857e1a70 /* 31 vars */) = 0
brk(NULL) = 0x55b68aefa000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc59a23cb0) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
opentat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=31379, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 31379, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fa980ad9000
close(3) = 0
opentat(AT_FDCWD, "/lib64/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0Pp\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=175552, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fa980ad7000
mmap(NULL, 181896, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fa980aaa000
mmap(0x7fa980ab0000, 110592, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7fa980ab0000
mmap(0x7fa980acb000, 32768, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x21000) = 0x7fa980acb000
mmap(0x7fa980ad3000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7fa980ad3000
mmap(0x7fa980ad5000, 5768, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fa980ad5000
close(3) = 0
write(1, " echo.py\t\t\t gnu_compiler_optimi"..., 103
echo.py      gnu_compiler_optimisations      OpenMP-offload-training
sse0.cpp     sse2          vector_op.cpp
) = 103
close(1) = 0
close(2) = 0
exit_group(0) = ?
+++ exited with 0 +++
```



Guts of the program

# Tracing program failures

```
[matthew@moonshine ~]$ strace cat .bashrc
```

```
<snip>
```

```
read(3, "# .bashrc\n\n# Source global defin"..., 131072) = 765
```

```
<snip>
```

```
[matthew@moonshine ~]$ echo $?
```

```
0
```



# Tracing program failures

```
[matthew@moonshine ~]$ cat badfile  
cat: badfile: No such file or directory
```

# Tracing program failures

```
[matthew@moonshine ~]$ echo $?
```

```
1
```

```
[matthew@moonshine ~]$ echo $?
```

```
0
```

Any return code other than 0 indicates a failure.

Only one success value because there is only one way to do what was expected and many ways to fail.

# Tracing program failures

```
[matthew@moonshine ~]$ strace cat badfile
```

Any return code other than 0 indicates a failure.

Only one success value because there is only one way to do what was expected and many ways to fail.

# Tracing program failures

```
[matthew@moonshine ~]$ strace cat badfile
```

```
<snip>
```

```
  openat(AT_FDCWD, "badfile", O_RDONLY) = -1 ENOENT (No such file or directory)
```

```
<snip>
```

```
open(2)
```

```
System Calls Manual
```

**NAME**

open, openat, creat - open and possibly create a file

**LIBRARY**

Standard C library (libc, -lc)

# Viewing Threads

- The “m” flag

```
[matthew@moonshine]$ watch ps m
```

“Watch” repeats whatever command comes next every 2 seconds

“ps m” shows processes and their associated threads

(htop showed the thread PIDs, “ps m” shows only parent thread with a dash for each thread.)

# Add "m" to top and ps

```
[matthew@moonshine FORTRAN_SMP]$ time OMP_NUM_THREADS=8 ./smp_vecadd 1000000000
```

```
real 0m7.459s  
user 0m10.917s  
sys 0m3.000s
```

2.0s: ps m

moonshine: Thu Feb

ID	TTY	STAT	TIME	COMMAND
05	pts/0	-	0:00	-bash
-	-	Ss+	0:00	-
23	pts/1	-	0:00	-bash
-	-	Ss	0:00	-
59	pts/3	-	0:00	-bash
-	-	Ss	0:00	-
50	pts/3	-	0:00	watch ps m
-	-	S+	0:00	-
65	pts/1	-	0:07	./smp_vecadd 1000000000
-	-	Rl+	0:06	-
-	-	Rl+	0:00	-
-	-	Rl+	0:00	-
-	-	Rl+	0:00	-
-	-	Rl+	0:00	-
-	-	Rl+	0:00	-
-	-	Rl+	0:00	-
-	-	Rl+	0:00	-
71	pts/3	-	0:00	watch ps m
-	-	S+	0:00	-
72	pts/3	-	0:00	ps m
-	-	R+	0:00	-



# Resource Monitoring

- Time
- Top
- Uptime

# System Load

```
[matthew@moonshine FORTRAN_SMP]$ top
```

```
top - 09:10:07 up 12 days, 19:35, 2 users, load average: 0.03, 0.02, 0.00
Tasks: 420 total, 1 running, 419 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 63774.3 total, 61049.6 free, 1551.6 used, 1993.5 buff/cache
MiB Swap: 32208.0 total, 32208.0 free, 0.0 used. 62222.7 avail Mem
```

Load over the past minute: 0.03

5 mins: 0.02

15 mins 0.00

The load is the average number of processes ready to execute.

That's the number of running processes plus the number ready for the CPU.

It's the number of processes not waiting for anything other than CPU time.

# System Load

```
[matthew@moonshine FORTRAN_SMP]$ top
```

```
top - 09:10:07 up 12 days, 19:35, 2 users, load average: 0.03, 0.02, 0.00
Tasks: 420 total, 1 running, 419 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 63774.3 total, 61049.6 free, 1551.6 used, 1993.5 buff/cache
MiB Swap: 32208.0 total, 32208.0 free, 0.0 used. 62222.7 avail Mem
```

Load over the past minute: 0.03

5 mins: 0.02

15 mins 0.00

On a perfectly efficient system the load would be equal to the number of CPUs (cores). **Why not more? Why not less?**

# Memory usage

- Yum install time.
- This is going to be confusing. There is a "time" command built into the shell that you have been using.
- Now we have installed a "time" program. The shell command is going to shadow the program (the shell finds and runs it before checking the PATH variable for non-shell commands)
- So we have to run it explicitly with /usr/bin/time.

# Pagefaults (Start here on Wednesday)

```
[matthew@moonshine FORTRAN_SMP]$ export OMP_NUM_THREADS=8
[matthew@moonshine FORTRAN_SMP]$ /usr/bin/time ./smp_vecadd 1000000000
10.94user 3.01system 0:07.46elapsed 186%CPU (0avgtext+0avgdata 11721652maxresident)k
0inputs+0outputs (2major+9195minor)pagefaults 0swaps
```

The program had to talk to the memory management unit (kernel lecture) **9195** times to get physical RAM mapped to its VM space. This is because the memory was allocated on the **heap at runtime**.

Twice data had to be loaded from disk. (This was because the program had to be loaded into memory)

Try running the command again and the major faults should go away. Why?

# Pagefaults

```
[matthew@moonshine FORTRAN_SMP]$ vmstat 2
procs -----memory----- ---swap-- -----io----- -system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs  us  sy  id  wa  st
 1  0     0 62519124  4312 2037292    0    0     0     0    0    1   0   0   0 100   0   0
 0  0     0 62519124  4312 2037292    0    0     0     0    0 196 182   0   0 100   0   0
 0  0     0 62519124  4312 2037292    0    0     0     0    0 185 186   0   0 100   0   0
 0  0     0 62519124  4312 2037292    0    0     0     0    0 150 167   0   0 100   0   0
 0  0     0 62519124  4312 2037292    0    0     0     0    0 147 154   0   0 100   0   0
```

Us: user

Sys: kernel

Id: idle

Wa: waiting for IO

In one terminal run `vmstat 2` and in another run your `smp_vecadd` program

# Pagefaults

```
[matthew@moonshine ]$ sudo yum install sysstat
```

```
[matthew@moonshine ]$ iostat 2
```

# Process monitoring

```
[matthew@moonshine FORTRAN_SMP]$ ./smp_vecadd 1000000000 &
[1] 226888
[matthew@moonshine FORTRAN_SMP]$ pidstat -p $(pgrep smp) 1
Linux 5.14.0-362.18.1.el9_3.0.1.x86_64 (moonshine) 03/04/2024 _x86_64_ (32 CPU)

09:44:33 AM    UID          PID    %usr %system  %guest   %wait   %CPU   CPU  Command
09:44:34 AM    1000        226888    77.00  23.00   0.00    0.00  100.00   21  smp_vecadd
09:44:35 AM    1000        226888    77.00  23.00   0.00    0.00  100.00   21  smp_vecadd
09:44:36 AM    1000        226888   308.00  72.00   0.00    0.00  380.00   21  smp_vecadd
[1]+  Done                  ./smp_vecadd 1000000000
```

Why was the CPU usage 100% for the first two seconds and then 380%?

CPU ID