Linux For Embedded Systems

For Arabs

Course 102:

Understanding Linux

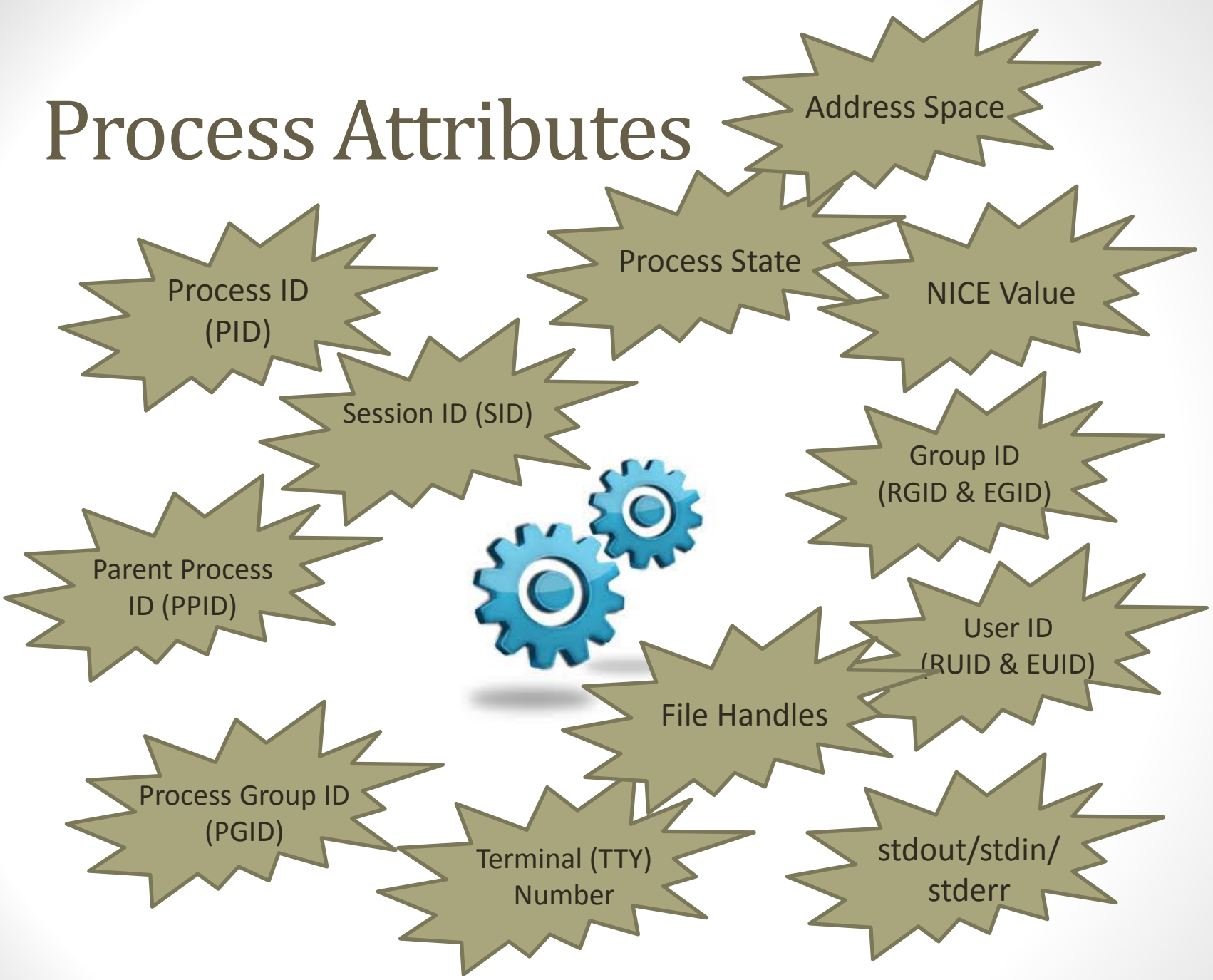
Ahmed ElArabawy

Linux4
Embedded Systems

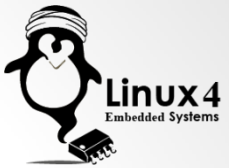
Lecture 17:
Process Monitoring

On a UNIX system, everything is a file; if something is not a file, it is a process

Process Attributes



Process Attributes

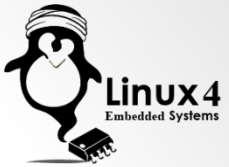


- Process Id (***PID***)
  - A unique identification number used to refer to the process
- Parent Process ID (***PPID***)
  - The process ID of the parent process
- Process Group ID (***PGID***)
  - The ID for the process group that this process belong to
  - Will be equal to **PID** if the process is a Process Group Leader
- Session ID (***SID***)
  - The ID for the session that this process belong to
  - The same for the **PID** if this process is the Session Leader
- Terminal Number
  - The number of the terminal that is associated with the process
  - Only applicable for interactive processes
  - Batch processes and Daemons have the value of 0

Process Attributes

- Owner User ID
  - There are two user IDs for the process
    - Real User ID (***RUID***): the user who started the process
    - Effective User ID (***EUID***): the user whose privileges will be inherited by the process when accessing resources
    - By default **RUID** and **EUID** has the same value
- Owner Group ID
  - Similar to user ID, there are two Group IDs
    - Real Group ID (***RGID***): The primary group of the user who started the process
    - Effective Group ID (***EGID***): The Group ID whose privileges are inherited by the process when accessing system resources
    - By default **RGID** and **EGID** has the same value

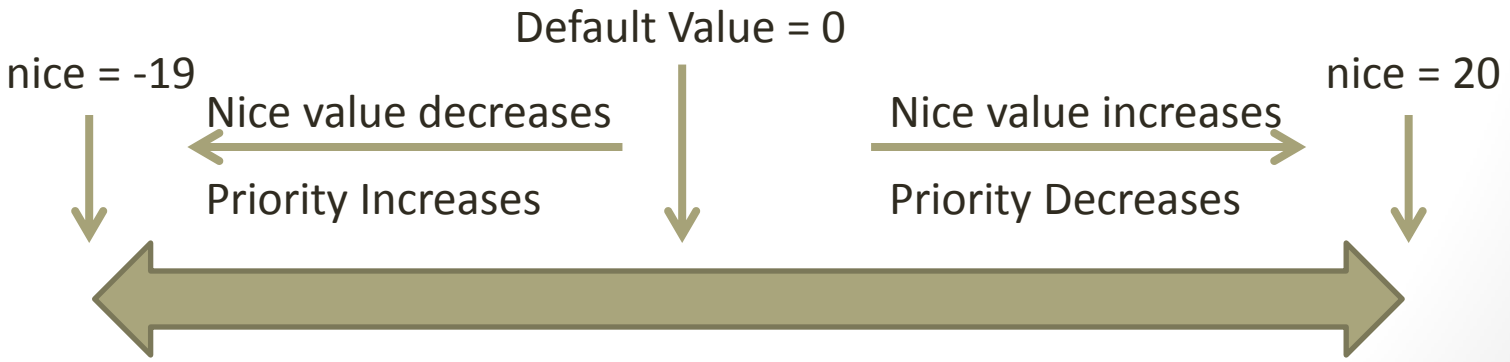
Process Priority



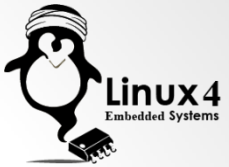
- Each process will have a priority value
- The process priority affects the scheduler behavior in the Linux Kernel
- A Process with higher priority may,
  - Receive more processor time by the kernel
  - Get scheduled more often
  - Preempt processes with lower processes
- Process priority is defined by,
  - User Controlled value (***nice*** Value)
  - Kernel calculated value (based on the process behavior)
    - Processing bound
    - I/O bound

The “nice” Value

- This is a value between -19 to 20 assigned by the user
- The kernel uses this value as a factor in calculating the process priority
- The higher the “nice” value of the process, the more it is accommodating other processes (less aggressive), which means lower priority



Set Process Priority (nice Command)



$ nice -<value> command

$ nice -n <value> command

This command sets the nice value to manipulate the process priority

- To increase priority, **nice** should be decreased and vice versa
  - **nice** value vary between -20 to +19
  - Default **nice** value is “0”
- Example:
  - To set **nice** to value 5 for a new process,

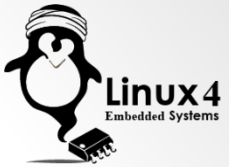
    $ nice -5 processCommand &

    $ nice -n 5 processCommand &
  - To set nice to value -10 for a new process

    $ nice --10 processCommand &

    $ nice -n -10 processCommand &
- Note: non-root users can only do 1-19 nice values

Modify Process Priority (renice Command)



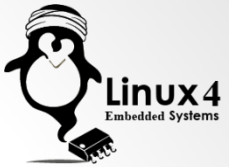
$ renice <value> <pid of process>

This command changes the "**nice**" value of a process

Example:

$ renice 10 1002

- Note,
  - Non-root users can only increase the processes **nice** value (decrease its priority)
  - The passed value is the new **nice** value and not a increment/decrement value

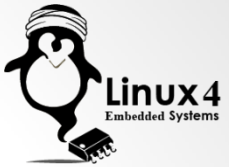


DISPLAY PROCESS INFORMATION & SYSTEM RESOURCE USAGE

Displaying Process Info

Command	Function
$ ps	Display Process Static Information
$ pstree	Display Process Hierarchy
$ top	Display Resource usage by processes
$ vmsat	Display System Resource Usage
$ xload ***$ tload***	System load average (in Graphical fashion)
$ free	Display of Memory Usage (used/free)
$ time	Measuring Running Time of a Command
$ uptime	Measuring System Uptime

Display Process Info (ps Command)

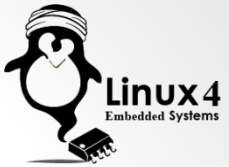


$ ps <options>

Display Information about the running processes

- The ***ps*** command is a very powerful and diverse command
- It has a lot of options to change the ways to display process info,
  - Some options are used to define the scope of which processes to show
  - Other options are used to define the format of the listing and which information to show
- we will go through the common forms for the command

The **ps** Command Defining Scope



$ ps

- Display info about processes under the current shell

$ ps -e

$ ps -A

- Display all processes in the system

$ ps a

- Display all processes in the system except those which are not attached with a terminal

$ ps ax

- Display all processes in the system including those which are not attached with a terminal

Defining Format

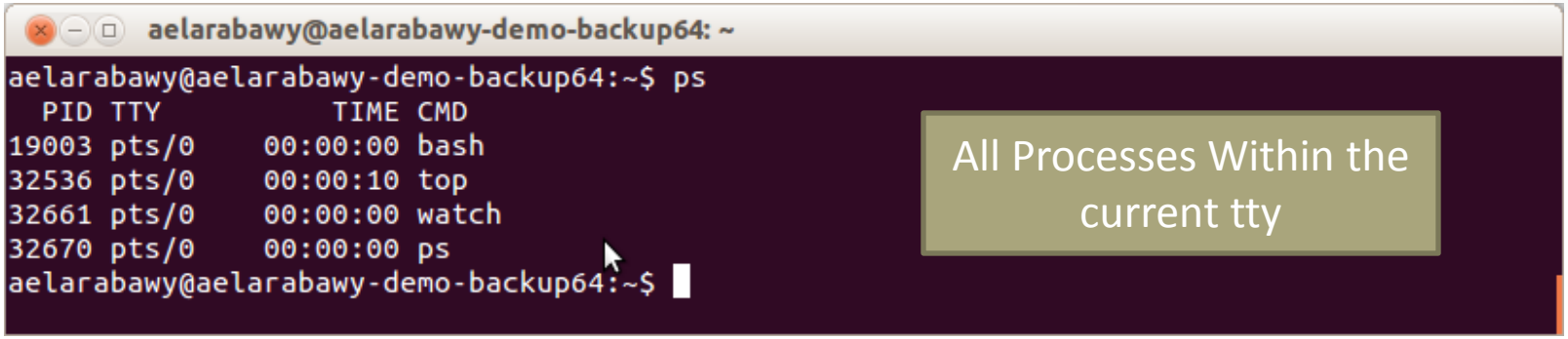
- You can use different options to define the format of the output (what columns to show)
- Some of the common usages

  $ ps -f

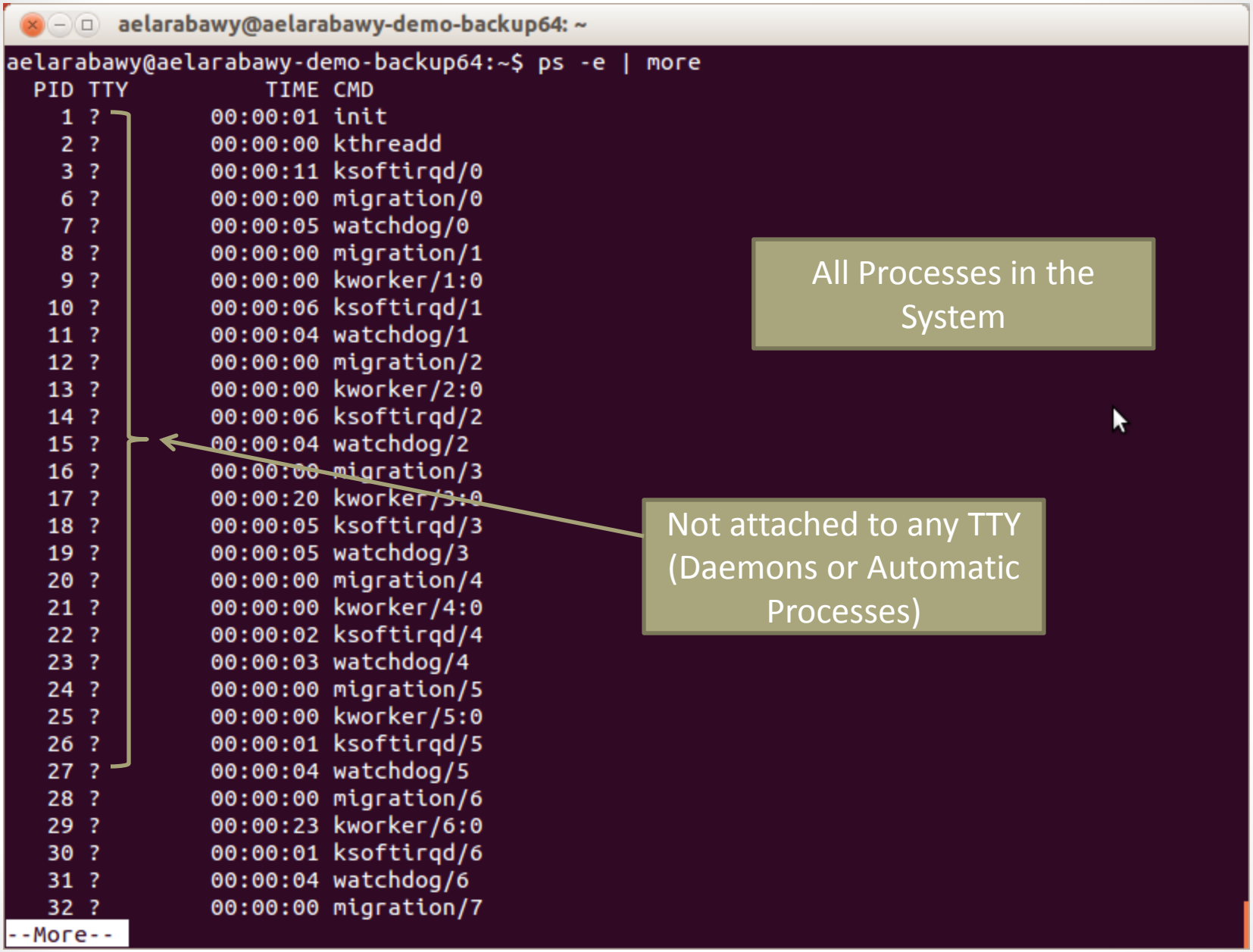
  $ ps -F

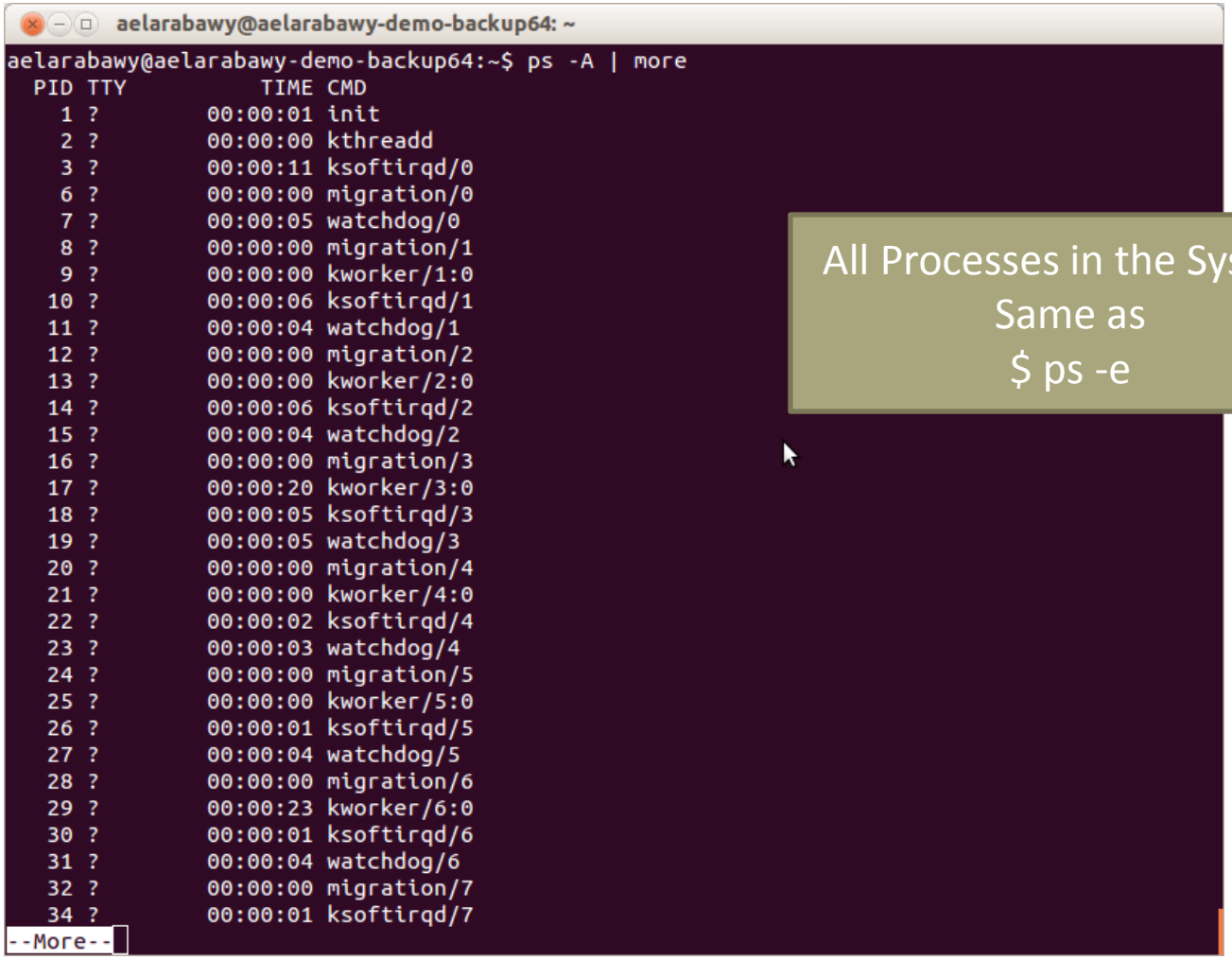
- To specify which fields to show use “-o” option

  $ ps -o pid,ppid,pgid,sid,command


aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ ps
PID TTY TIME CMD
19003 pts/0 00:00:00 bash
32536 pts/0 00:00:10 top
32661 pts/0 00:00:00 watch
32670 pts/0 00:00:00 ps
aelarabawy@aelarabawy-demo-backup64:~$
All Processes Within the current tty

```
prasad$ps
  PID TTY          TIME CMD
  659 pts/5    00:00:00 ksh
 1188 pts/5    00:00:00 ksh
 1543 pts/5    00:00:00 ksh
 1671 pts/5    00:00:00 ksh
 1676 pts/5    00:00:00 ksh
 8232 pts/5    00:00:00 ksh
 8380 pts/5    00:00:00 ksh
 8720 pts/5    00:00:00 ksh
 8904 pts/5    00:00:00 ksh
20882 pts/5    00:00:00 ksh
21116 pts/5    00:00:00 find
21903 pts/5    00:00:00 grep
22075 pts/5    00:00:00 ps
32726 pts/5    00:00:00 ksh
```


aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ ps -e | more
PID TTY TIME CMD
1 ? 00:00:01 init
2 ? 00:00:00 kthreadd
3 ? 00:00:11 ksoftirqd/0
6 ? 00:00:00 migration/0
7 ? 00:00:05 watchdog/0
8 ? 00:00:00 migration/1
9 ? 00:00:00 kworker/1:0
10 ? 00:00:06 ksoftirqd/1
11 ? 00:00:04 watchdog/1
12 ? 00:00:00 migration/2
13 ? 00:00:00 kworker/2:0
14 ? 00:00:06 ksoftirqd/2
15 ? 00:00:04 watchdog/2
16 ? 00:00:00 migration/3
17 ? 00:00:20 kworker/3:0
18 ? 00:00:05 ksoftirqd/3
19 ? 00:00:05 watchdog/3
20 ? 00:00:00 migration/4
21 ? 00:00:00 kworker/4:0
22 ? 00:00:02 ksoftirqd/4
23 ? 00:00:03 watchdog/4
24 ? 00:00:00 migration/5
25 ? 00:00:00 kworker/5:0
26 ? 00:00:01 ksoftirqd/5
27 ? 00:00:04 watchdog/5
28 ? 00:00:00 migration/6
29 ? 00:00:23 kworker/6:0
30 ? 00:00:01 ksoftirqd/6
31 ? 00:00:04 watchdog/6
32 ? 00:00:00 migration/7
--More--
All Processes in the System
Not attached to any TTY (Daemons or Automatic Processes)



All Processes in the System
Same as
$ ps -e

$ ps au

a : don't limit to same tty
but have to be attached to a tty

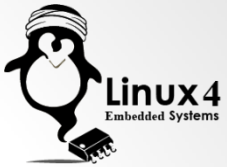
u : Long Format (more fields)

```
aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ ps au
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root       1111  0.0  0.0  19988   984 tty4     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty4
root       1116  0.0  0.0  19988   984 tty5     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty5
root       1123  0.0  0.0  19988   972 tty2     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty2
root       1125  0.0  0.0  19988   980 tty3     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty3
root       1128  0.0  0.0  19988   976 tty6     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty6
root       1262  0.1  3.0 171268 246044 tty7    Ss+  Jun03  64:02 /usr/bin/Xorg :0 -backgrou
root       1498  0.0  0.0  19988   984 tty1     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty1
1001      19003  0.0  0.0  28788  5904 pts/0    Ss   Jun23   0:00 bash
1001      19063  0.0  0.0  28744  5864 pts/1    Ss+  Jun23   0:00 bash
1001      30658  0.0  0.0  28724  5776 pts/2    Ss+  Jul01   0:00 bash
1001      32536  0.1  0.0  17456  1528 pts/0    T    09:03   0:10 top
1001      32661  0.0  0.0  18016  1440 pts/0    T    11:18   0:00 watch -n 2 ls
1001      32727  0.0  0.0  22360  1280 pts/0    R+   11:27   0:00 ps au
aelarabawy@aelarabawy-demo-backup64:~$
```

Keep in mind that some options are preceded with "-" and others not

For example:

$ ps -a is not the same as $ ps a



$ ps aux

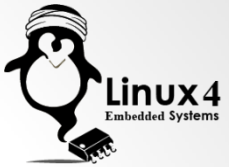
a : don’t limit to same tty
but have to be attached to a tty

x : Remove the tty restriction

u : Long Format (more fields)

aelarabawy@aelarabawy-demo-backup64: ~

```
aelarabawy@aelarabawy-demo-backup64:~$ ps aux | more
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.0  24732  2692 ?        Ss   Jun03   0:01 /sbin/init
root          2  0.0  0.0      0     0 ?        S    Jun03   0:00 [kthreadd]
root          3  0.0  0.0      0     0 ?        S    Jun03   0:11 [ksoftirqd/0]
root          6  0.0  0.0      0     0 ?        S    Jun03   0:00 [migration/0]
root          7  0.0  0.0      0     0 ?        S    Jun03   0:05 [watchdog/0]
root          8  0.0  0.0      0     0 ?        S    Jun03   0:00 [migration/1]
root          9  0.0  0.0      0     0 ?        S    Jun03   0:00 [kworker/1:0]
root         10  0.0  0.0      0     0 ?        S    Jun03   0:06 [ksoftirqd/1]
root         11  0.0  0.0      0     0 ?        S    Jun03   0:04 [watchdog/1]
root         12  0.0  0.0      0     0 ?        S    Jun03   0:00 [migration/2]
root         13  0.0  0.0      0     0 ?        S    Jun03   0:00 [kworker/2:0]
root         14  0.0  0.0      0     0 ?        S    Jun03   0:06 [ksoftirqd/2]
root         15  0.0  0.0      0     0 ?        S    Jun03   0:04 [watchdog/2]
```

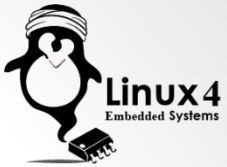


$ ps a -F

a : don't limit to same tty
but have to be attached to a tty

-F : Long Format (more fields)

```
aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ ps a -F
UID        PID  PPID  C    SZ   RSS PSR STIME TTY      STAT   TIME CMD
root      1111     1  0  4997   984   3 Jun03 tty4     Ss+    0:00 /sbin/getty -8 38400 tty4
root      1116     1  0  4997   984   3 Jun03 tty5     Ss+    0:00 /sbin/getty -8 38400 tty5
root      1123     1  0  4997   972   2 Jun03 tty2     Ss+    0:00 /sbin/getty -8 38400 tty2
root      1125     1  0  4997   980   3 Jun03 tty3     Ss+    0:00 /sbin/getty -8 38400 tty3
root      1128     1  0  4997   976   2 Jun03 tty6     Ss+    0:00 /sbin/getty -8 38400 tty6
root      1262  1184  0 42817 251924  0 Jun03 tty7     Ss+   64:06 /usr/bin/Xorg :0 -background
root      1498     1  0  4997   984   3 Jun03 tty1     Ss+    0:00 /sbin/getty -8 38400 tty1
1001     19003 18996  0  7197  5904   1 Jun23 pts/0    Ss     0:00 bash
1001     19063 18996  0  7186  5864   0 Jun23 pts/1    Ss+    0:00 bash
1001     30658 18996  0  7181  5776   0 Jul01 pts/2    Ss+    0:00 bash
1001     32536 19003  0  4364  1528   7 09:03 pts/0    T      0:10 top
1001     32661 19003  0  4504  1440   6 11:18 pts/0    T      0:00 watch -n 2 ls
1001     32767 19003  0  5590  1284   0 11:31 pts/0    R+     0:00 ps a -F
aelarabawy@aelarabawy-demo-backup64:~$
```



$ ps a -f

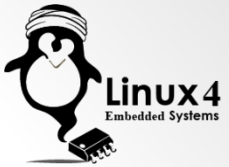
a : don't limit to same tty

but have to be attached to a tty

-f : Long Format (more fields)

aelarabawy@aelarabawy-demo-backup64: ~

```
aelarabawy@aelarabawy-demo-backup64:~$ ps a -f
UID        PID  PPID  C STIME TTY      STAT   TIME CMD
1001       308 19003  0 11:31 pts/0    R+     0:00 ps a -f
root      1111     1  0 Jun03 tty4     Ss+    0:00 /sbin/getty -8 38400 tty4
root      1116     1  0 Jun03 tty5     Ss+    0:00 /sbin/getty -8 38400 tty5
root      1123     1  0 Jun03 tty2     Ss+    0:00 /sbin/getty -8 38400 tty2
root      1125     1  0 Jun03 tty3     Ss+    0:00 /sbin/getty -8 38400 tty3
root      1128     1  0 Jun03 tty6     Ss+    0:00 /sbin/getty -8 38400 tty6
root      1262  1184  0 Jun03 tty7     Ss+   64:06 /usr/bin/Xorg :0 -background none -verbose -
root      1498     1  0 Jun03 tty1     Ss+    0:00 /sbin/getty -8 38400 tty1
1001     19003 18996  0 Jun23 pts/0    Ss     0:00 bash
1001     19063 18996  0 Jun23 pts/1    Ss+    0:00 bash
1001     30658 18996  0 Jul01 pts/2    Ss+    0:00 bash
1001     32536 19003  0 09:03 pts/0    T      0:10 top
1001     32661 19003  0 11:18 pts/0    T      0:00 watch -n 2 ls
aelarabawy@aelarabawy-demo-backup64:~$
```



$ ps auf

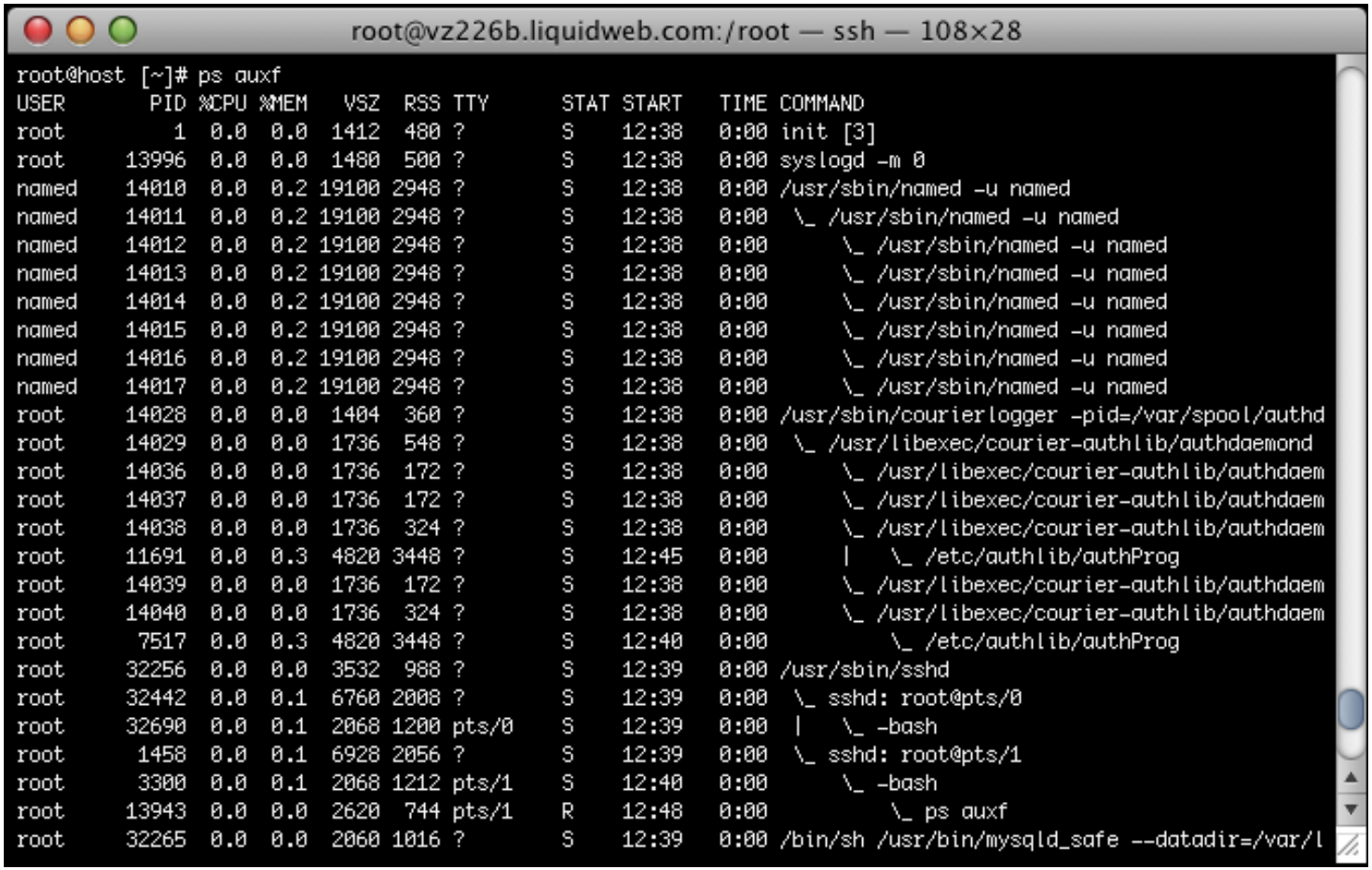
a : don't limit to same tty
but have to be attached to a tty

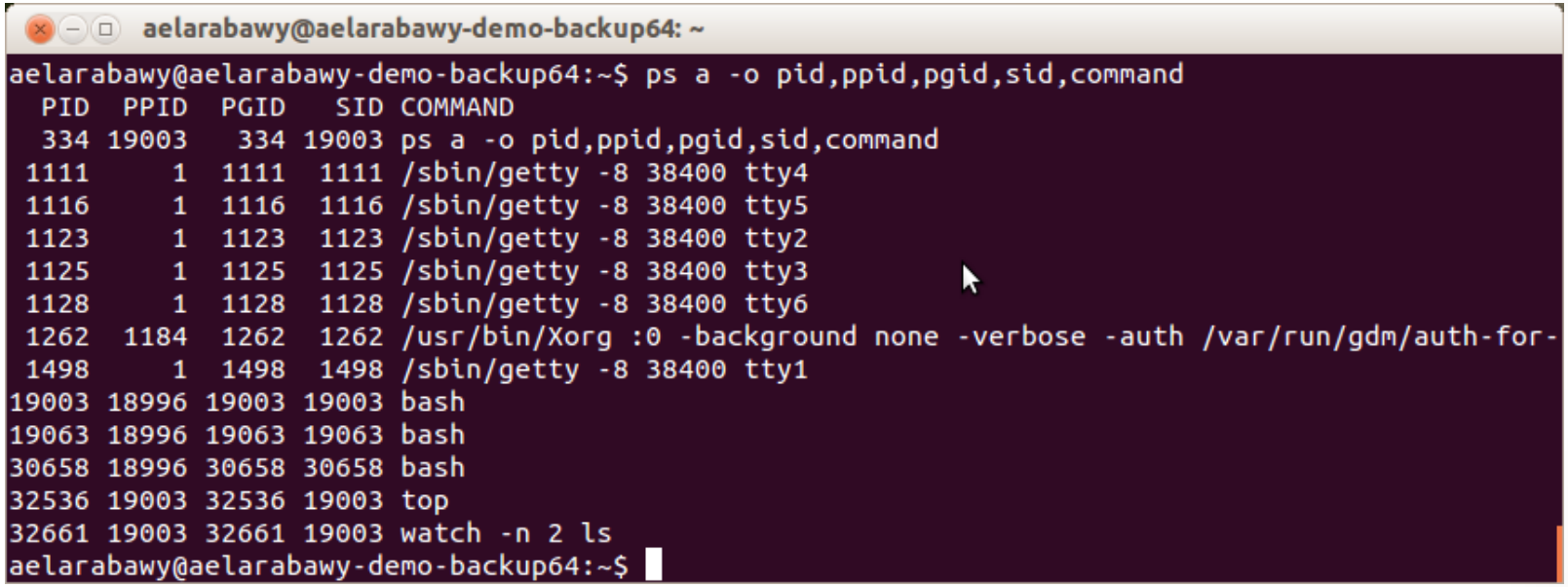
u : Long Format

f : Show process in tree format

aelarabawy@aelarabawy-demo-backup64: ~

```
aelarabawy@aelarabawy-demo-backup64:~$ ps auf
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
1001     30658  0.0  0.0  28724  5776 pts/2    Ss+  Jul01   0:00 bash
1001     19063  0.0  0.0  28744  5864 pts/1    Ss+  Jun23   0:00 bash
1001     19003  0.0  0.0  28788  5904 pts/0    Ss   Jun23   0:00 bash
1001     32536  0.0  0.0  17456  1528 pts/0    T    09:03   0:10  \_ top
1001     32661  0.0  0.0  18016  1440 pts/0    T    11:18   0:00  \_ watch -n 2 ls
1001       378  0.0  0.0  22328  1144 pts/0    R+   12:07   0:00  \_ ps auf
root      1262  0.1  3.0 171268 245684 tty7    Ss+  Jun03  64:14 /usr/bin/Xorg :0 -background n
root      1498  0.0  0.0  19988   984 tty1     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty1
root      1128  0.0  0.0  19988   976 tty6     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty6
root      1125  0.0  0.0  19988   980 tty3     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty3
root      1123  0.0  0.0  19988   972 tty2     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty2
root      1116  0.0  0.0  19988   984 tty5     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty5
root      1111  0.0  0.0  19988   984 tty4     Ss+  Jun03   0:00 /sbin/getty -8 38400 tty4
aelarabawy@aelarabawy-demo-backup64:~$
```


root@vz226b.liquidweb.com:/root — ssh — 108×28
root@host [~]# ps auxf
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.0 1412 480 ? S 12:38 0:00 init [3]
root 13996 0.0 0.0 1480 500 ? S 12:38 0:00 syslogd -m 0
named 14010 0.0 0.2 19100 2948 ? S 12:38 0:00 /usr/sbin/named -u named
named 14011 0.0 0.2 19100 2948 ? S 12:38 0:00 _ /usr/sbin/named -u named
named 14012 0.0 0.2 19100 2948 ? S 12:38 0:00 _ /usr/sbin/named -u named
named 14013 0.0 0.2 19100 2948 ? S 12:38 0:00 _ /usr/sbin/named -u named
named 14014 0.0 0.2 19100 2948 ? S 12:38 0:00 _ /usr/sbin/named -u named
named 14015 0.0 0.2 19100 2948 ? S 12:38 0:00 _ /usr/sbin/named -u named
named 14016 0.0 0.2 19100 2948 ? S 12:38 0:00 _ /usr/sbin/named -u named
named 14017 0.0 0.2 19100 2948 ? S 12:38 0:00 _ /usr/sbin/named -u named
root 14028 0.0 0.0 1404 360 ? S 12:38 0:00 /usr/sbin/courierlogger -pid=/var/spool/authd
root 14029 0.0 0.0 1736 548 ? S 12:38 0:00 _ /usr/libexec/courier-authlib/authdaemond
root 14036 0.0 0.0 1736 172 ? S 12:38 0:00 _ /usr/libexec/courier-authlib/authdaem
root 14037 0.0 0.0 1736 172 ? S 12:38 0:00 _ /usr/libexec/courier-authlib/authdaem
root 14038 0.0 0.0 1736 324 ? S 12:38 0:00 _ /usr/libexec/courier-authlib/authdaem
root 11691 0.0 0.3 4820 3448 ? S 12:45 0:00 | _ /etc/authlib/authProg
root 14039 0.0 0.0 1736 172 ? S 12:38 0:00 _ /usr/libexec/courier-authlib/authdaem
root 14040 0.0 0.0 1736 324 ? S 12:38 0:00 _ /usr/libexec/courier-authlib/authdaem
root 7517 0.0 0.3 4820 3448 ? S 12:40 0:00 _ /etc/authlib/authProg
root 32256 0.0 0.0 3532 988 ? S 12:39 0:00 /usr/sbin/sshd
root 32442 0.0 0.1 6760 2008 ? S 12:39 0:00 _ sshd: root@pts/0
root 32690 0.0 0.1 2068 1200 pts/0 S 12:39 0:00 | _ -bash
root 1458 0.0 0.1 6928 2056 ? S 12:39 0:00 _ sshd: root@pts/1
root 3300 0.0 0.1 2068 1212 pts/1 S 12:40 0:00 _ -bash
root 13943 0.0 0.0 2620 744 pts/1 R 12:48 0:00 _ ps auxf
root 32265 0.0 0.0 2060 1016 ? S 12:39 0:00 /bin/sh /usr/bin/mysqld_safe --datadir=/var/l


aelarabawy@aelarabawy-demo-backup64: ~
aelarabawy@aelarabawy-demo-backup64:~$ ps a -o pid,ppid,pgid,sid,command
PID PPID PGID SID COMMAND
334 19003 334 19003 ps a -o pid,ppid,pgid,sid,command
1111 1 1111 1111 /sbin/getty -8 38400 tty4
1116 1 1116 1116 /sbin/getty -8 38400 tty5
1123 1 1123 1123 /sbin/getty -8 38400 tty2
1125 1 1125 1125 /sbin/getty -8 38400 tty3
1128 1 1128 1128 /sbin/getty -8 38400 tty6
1262 1184 1262 1262 /usr/bin/Xorg :0 -background none -verbose -auth /var/run/gdm/auth-for-
1498 1 1498 1498 /sbin/getty -8 38400 tty1
19003 18996 19003 19003 bash
19063 18996 19063 19063 bash
30658 18996 30658 30658 bash
32536 19003 32536 19003 top
32661 19003 32661 19003 watch -n 2 ls
aelarabawy@aelarabawy-demo-backup64:~$

Showing Threads

- To show threads for multithreaded applications, use one of the following options

$ ps -H

$ ps -L

$ ps -T

$ ps -m

Common Practices

- There are hundreds of processes running in the system
- A common practice, is to do a full listing and pipe the output to a search filter to limit the list to what we are interested in

  $ ps -ef | grep root

  - This shows all processes owned by the root user

  $ ps aux | grep "pts/2"

  - This shows all processes attached to the tty pts/2

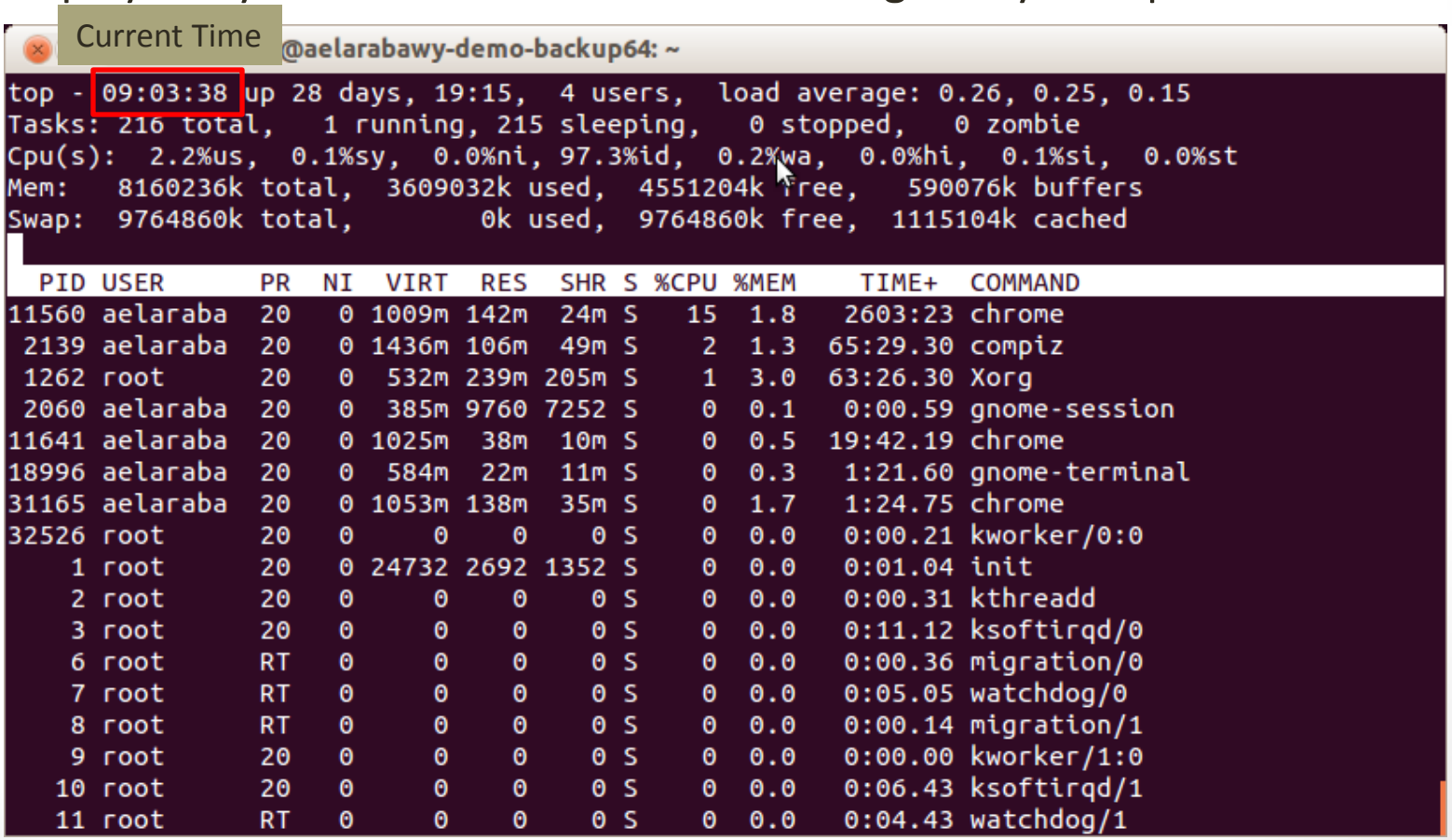
- There are a lot of options for scope and format. For a special need, it is advised to refer to the man page for the command

  $ man ps

Display Process Resource Usage (top Command)

$ top

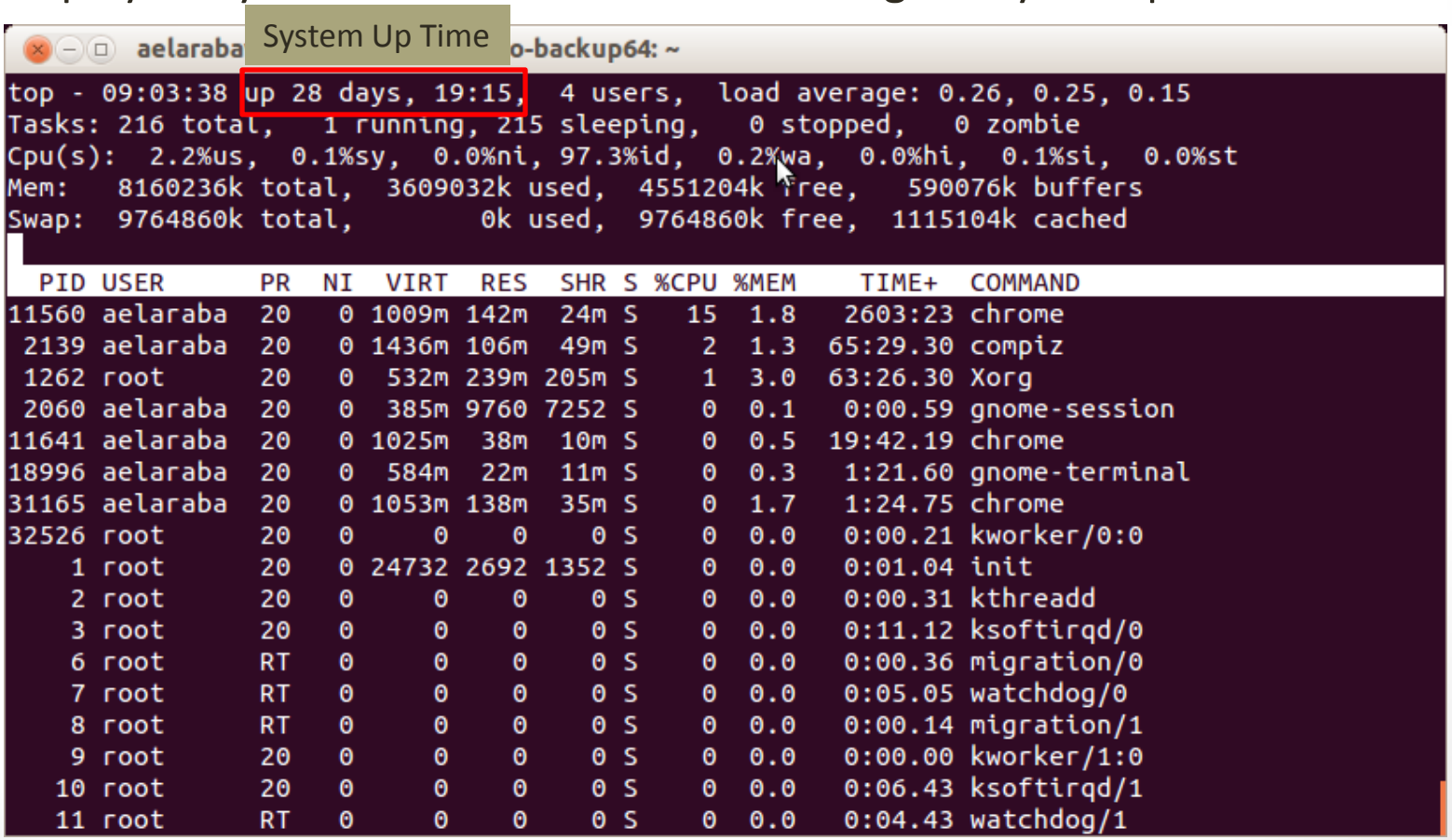
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

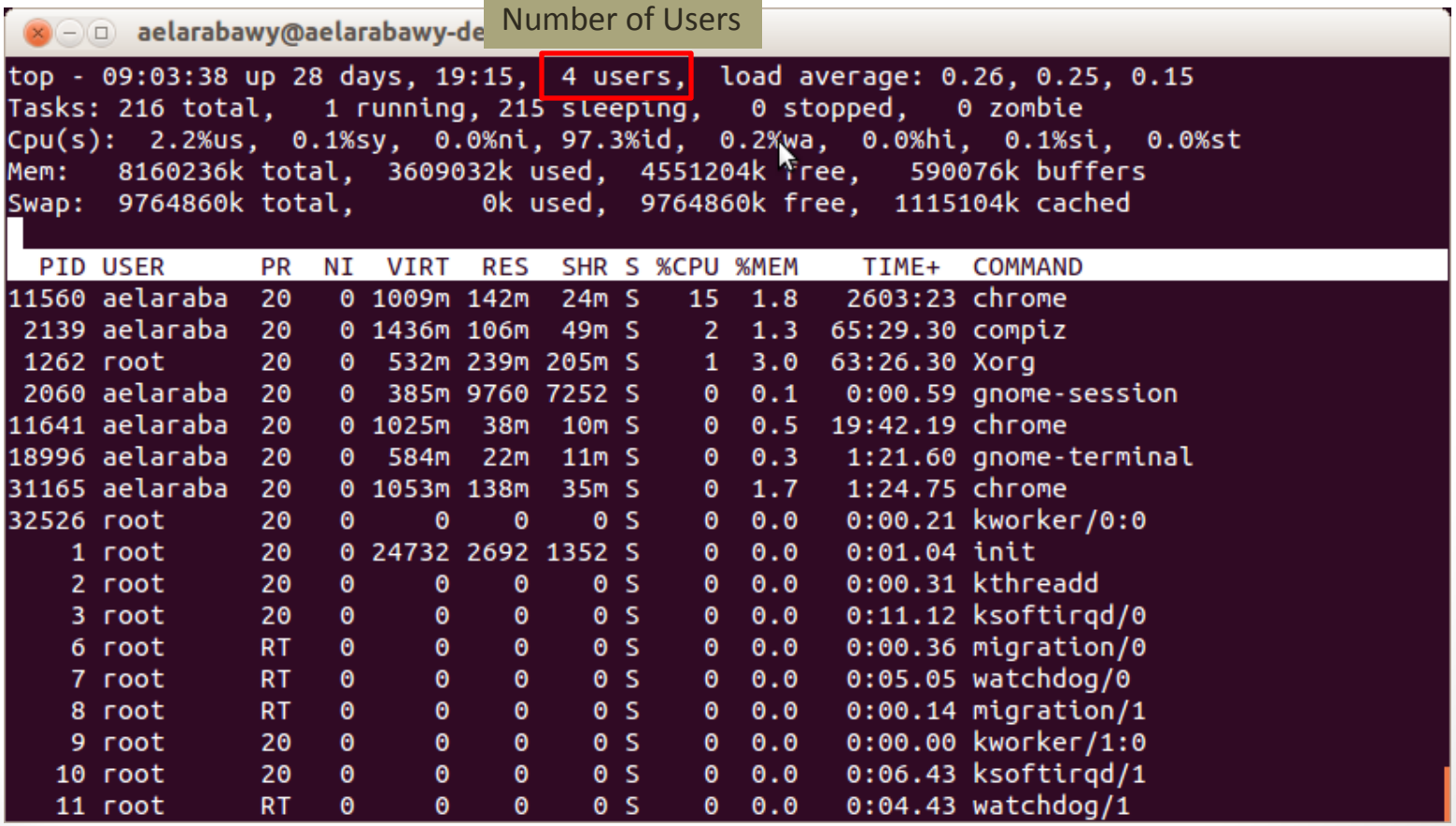
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

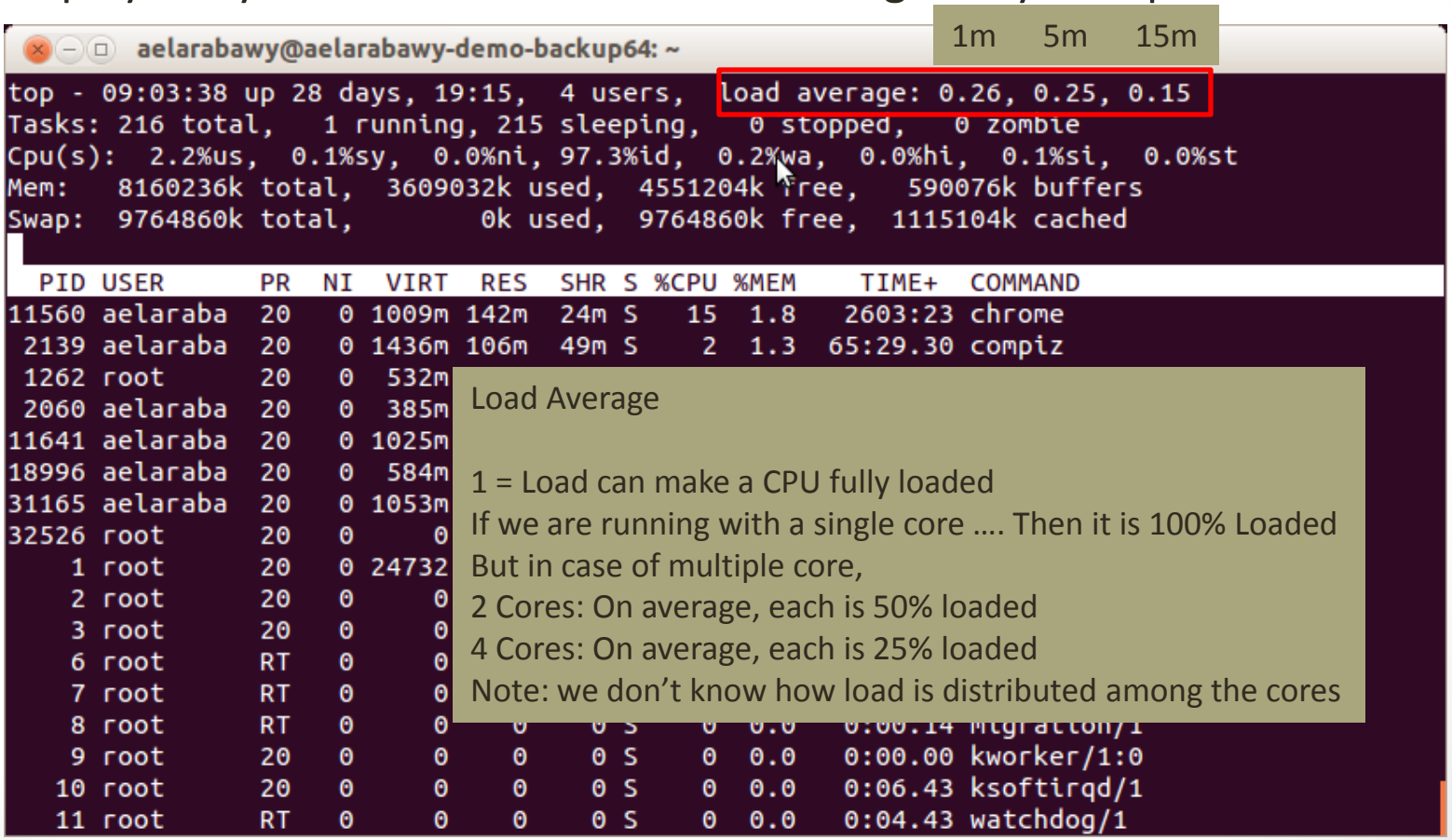
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

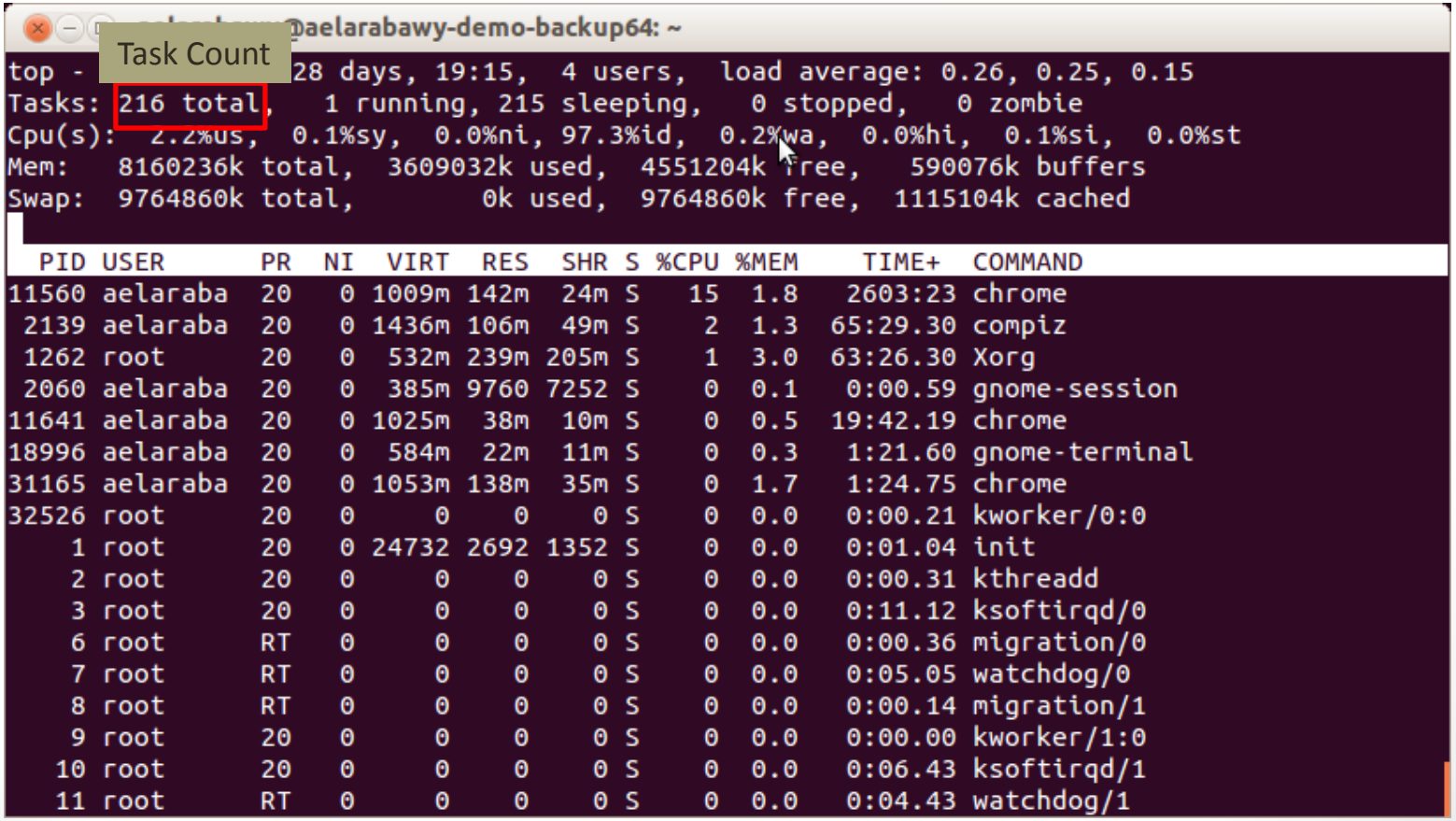
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

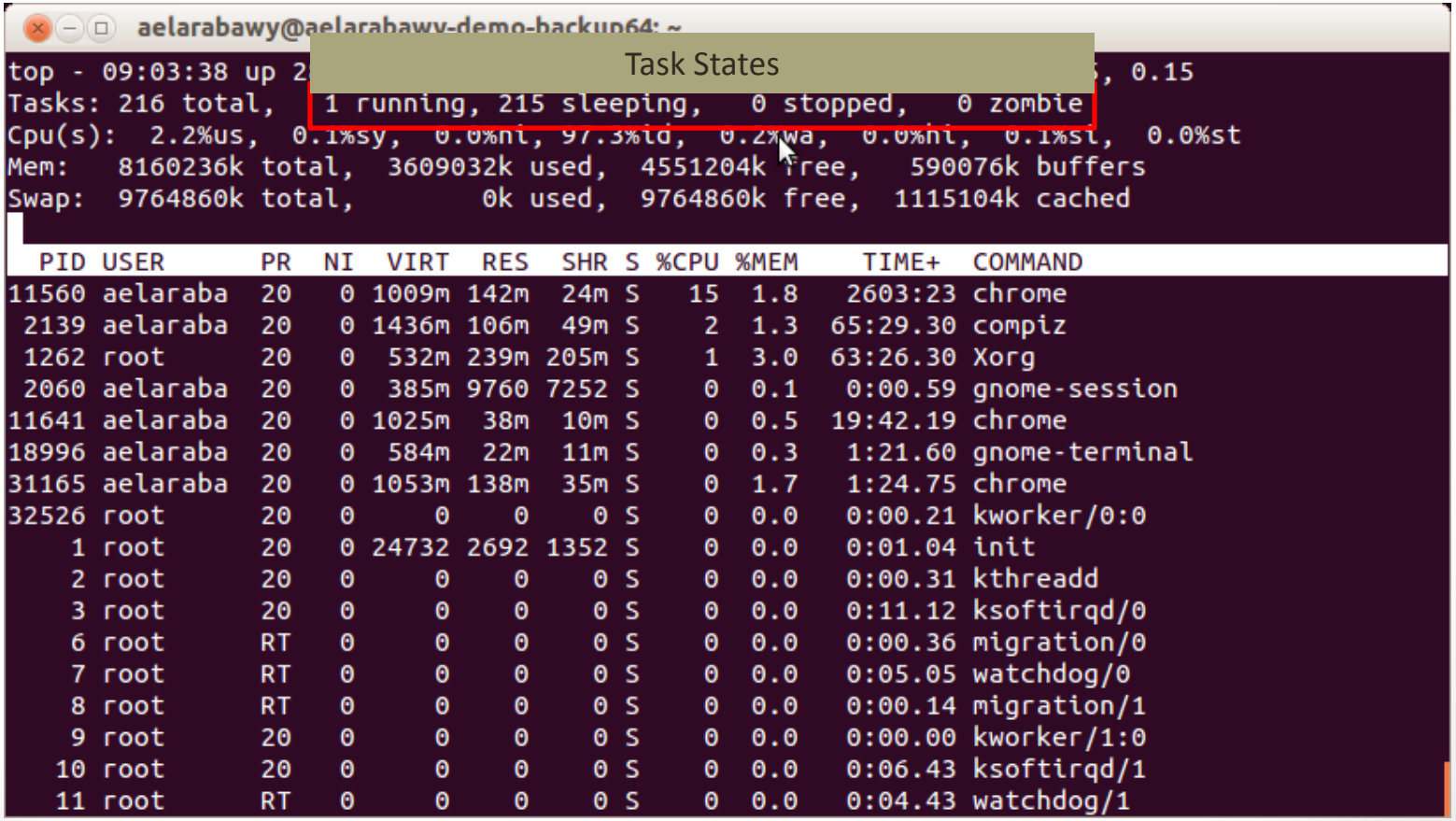
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

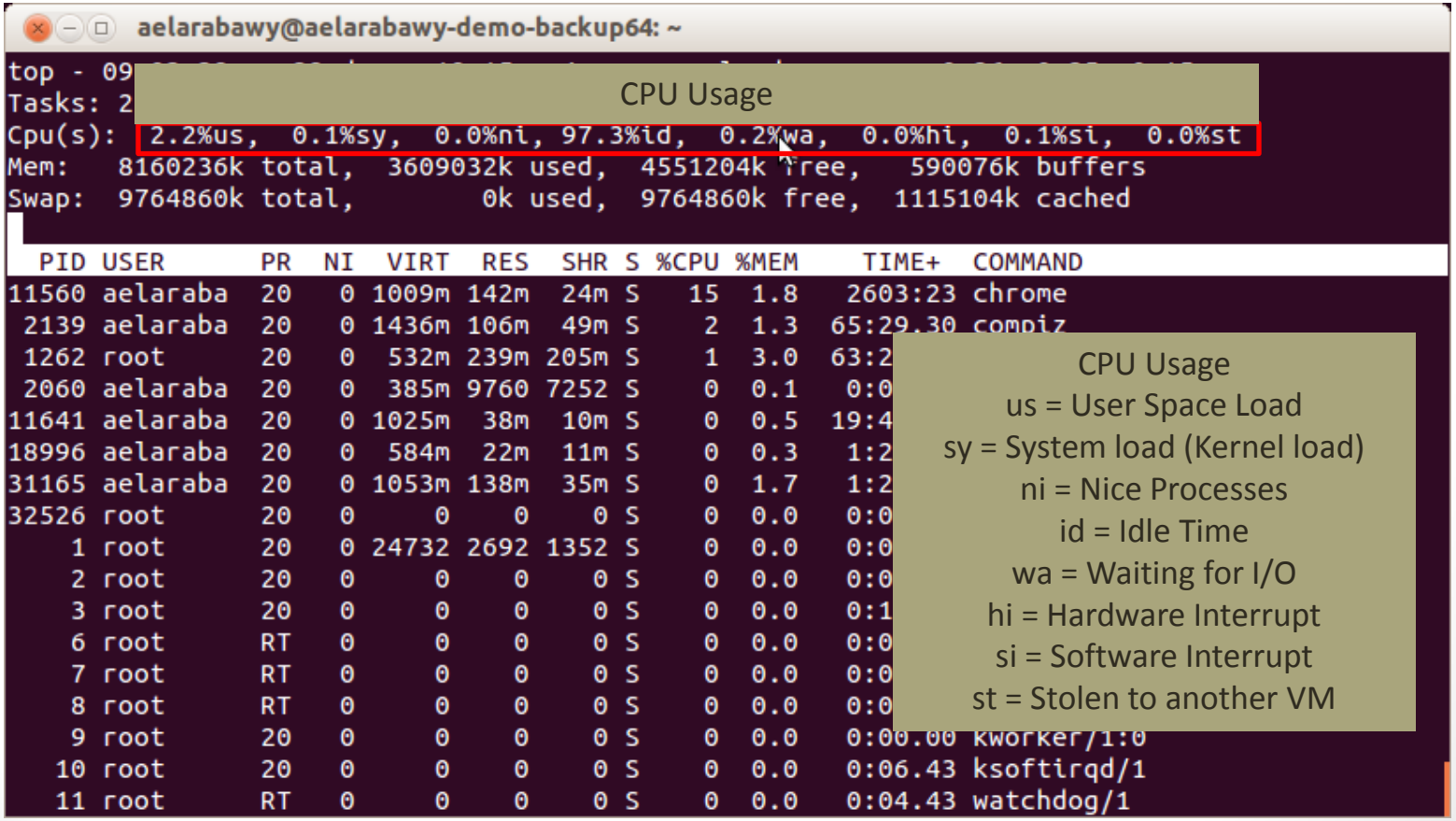
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

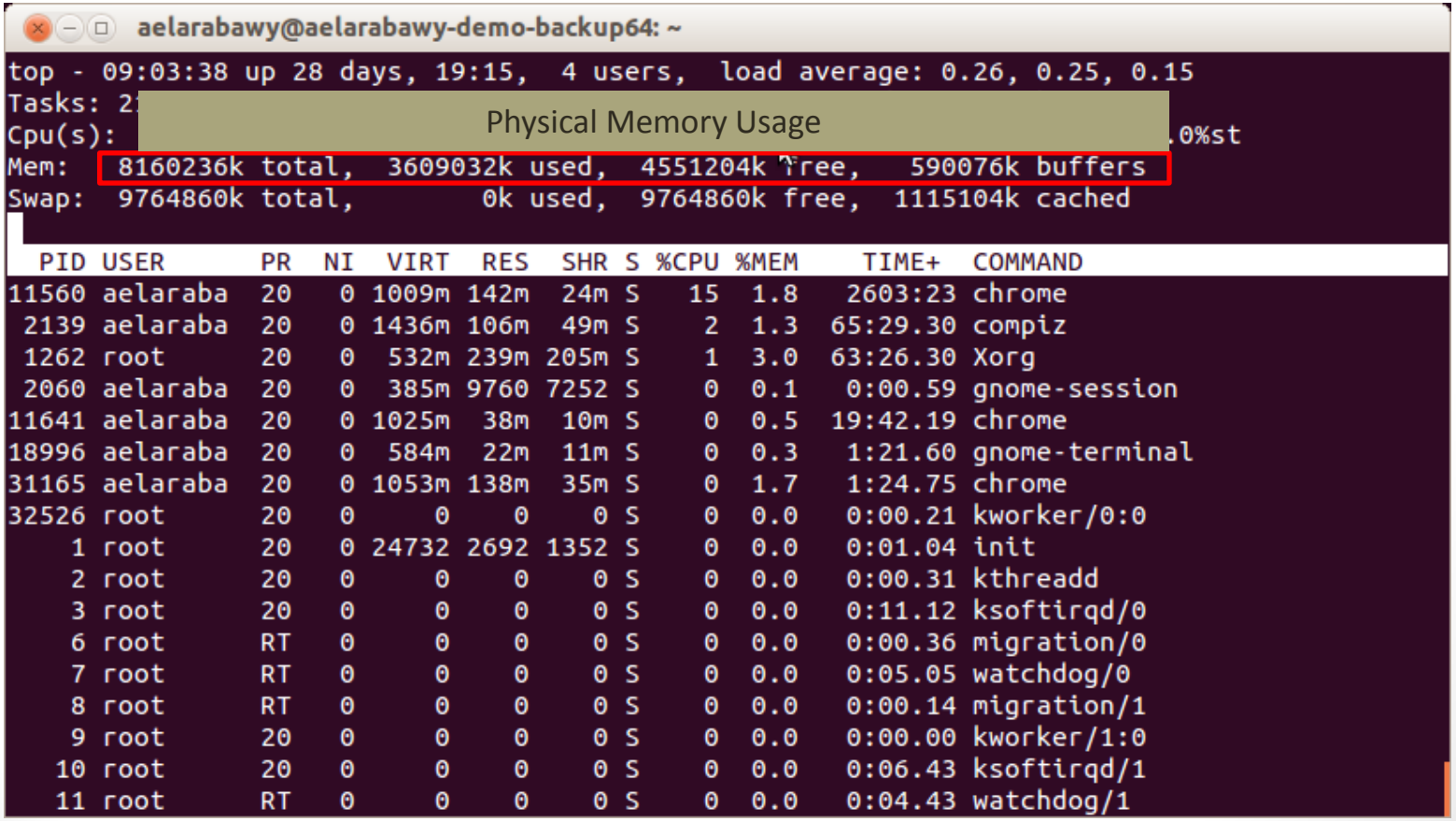
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

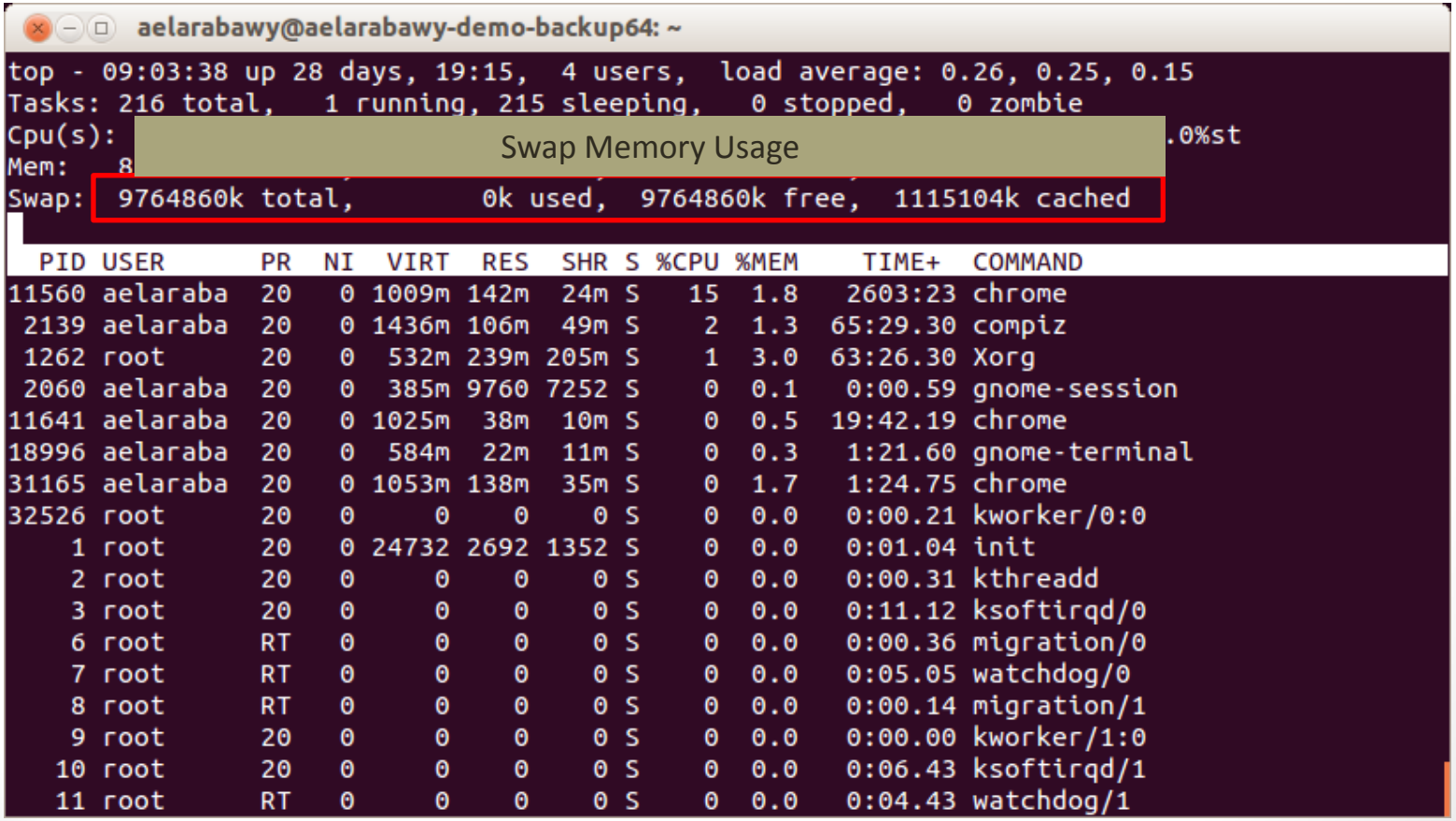
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

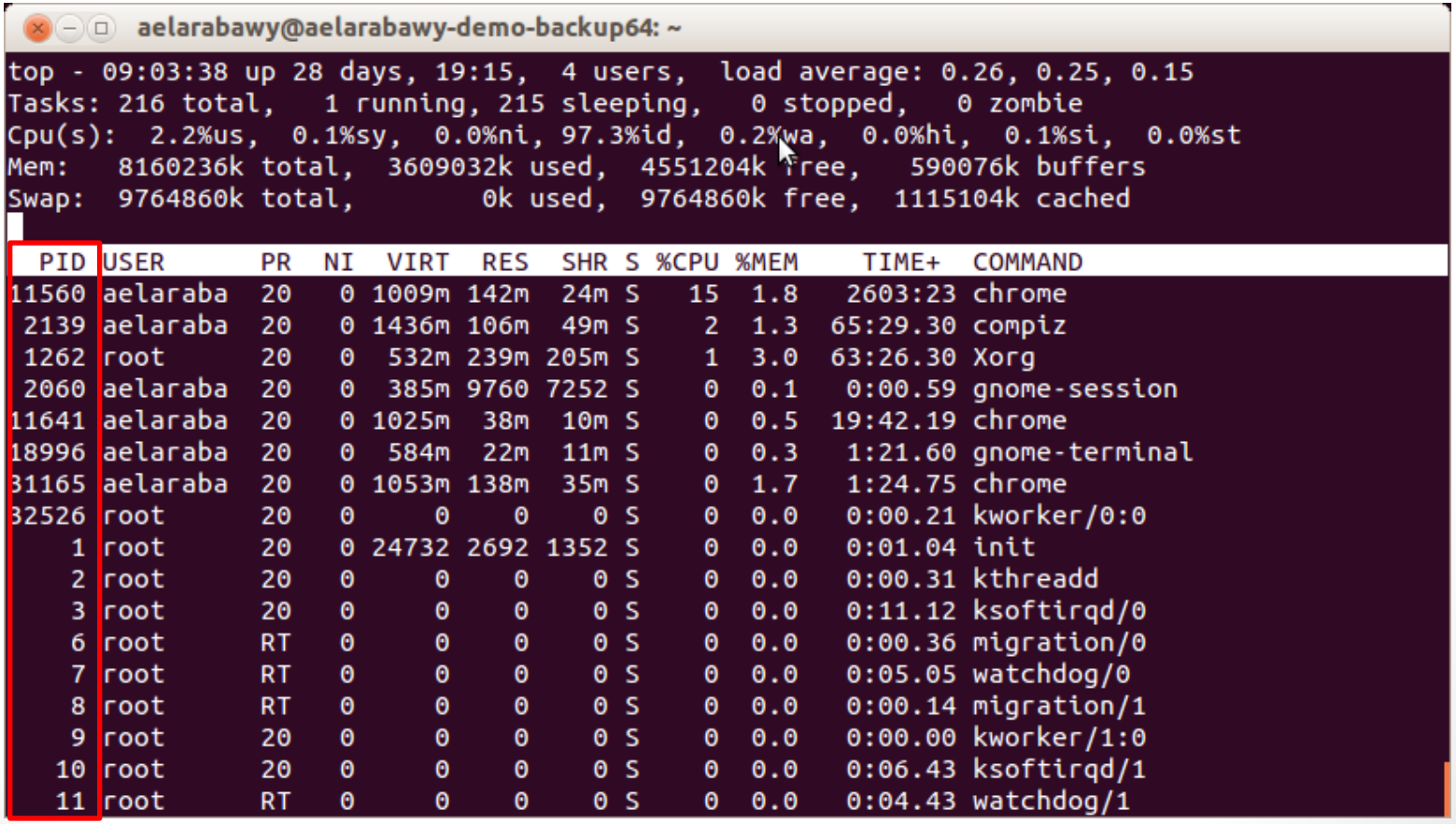
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

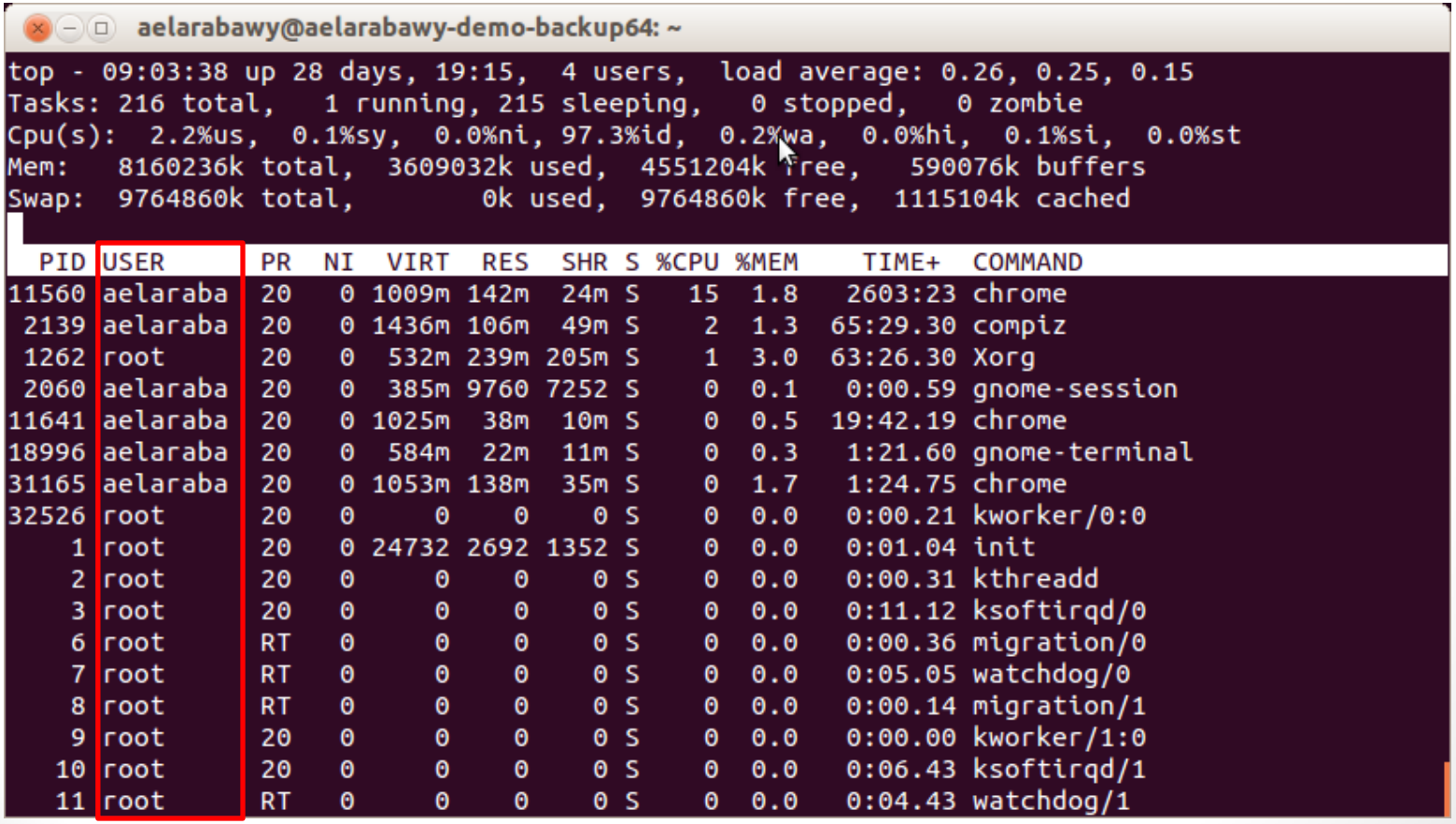
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

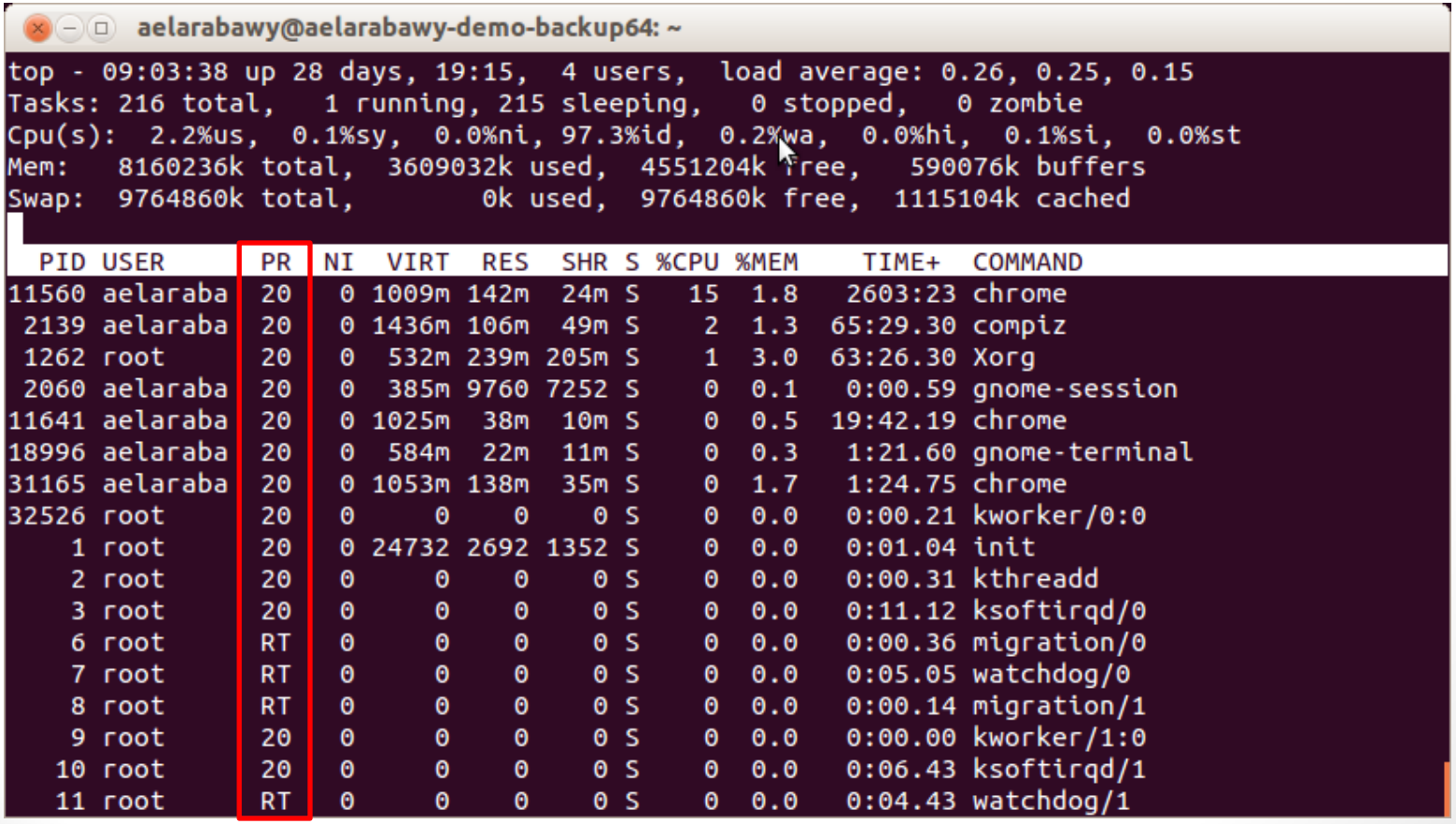
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

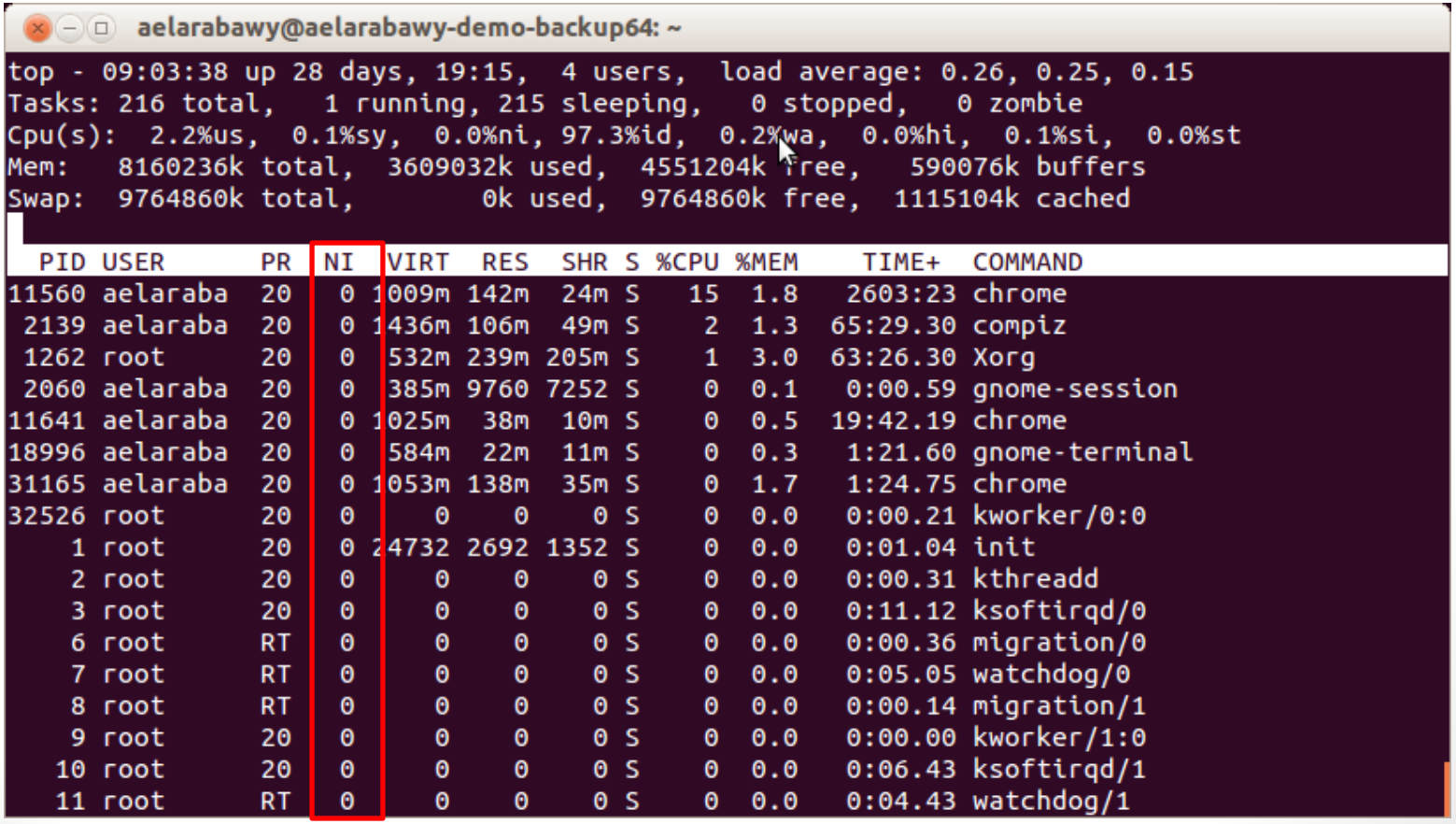
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

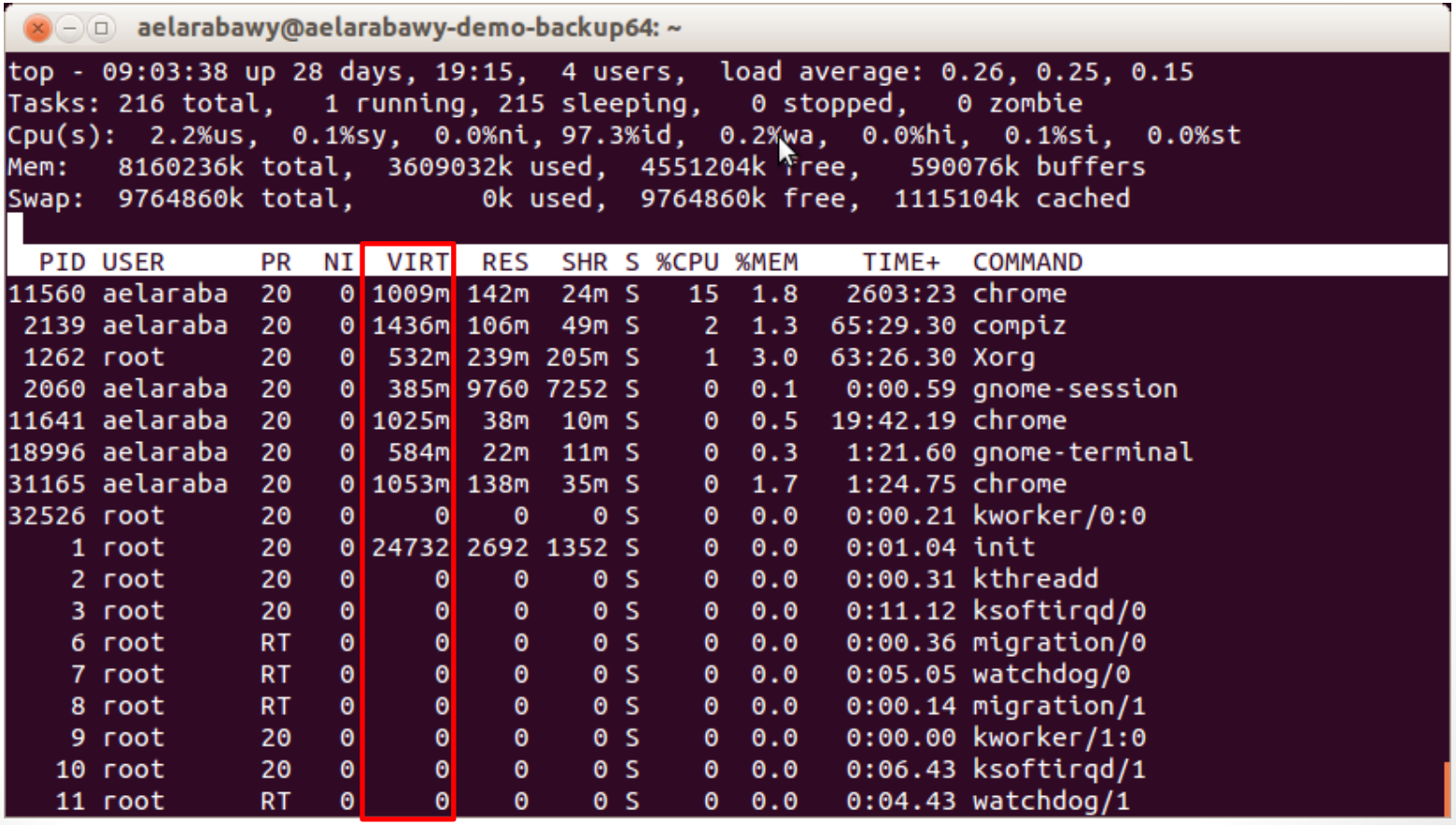
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

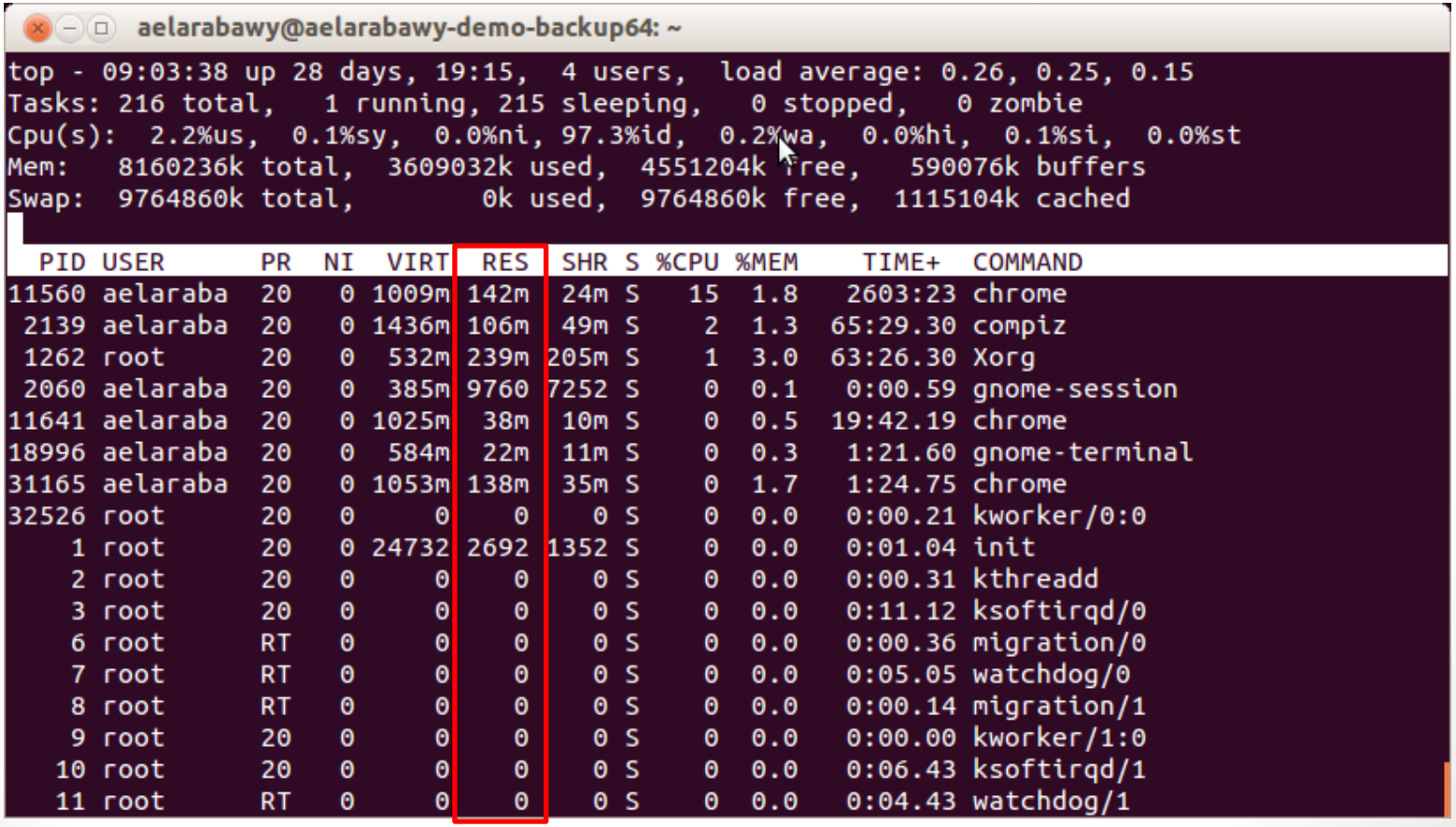
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

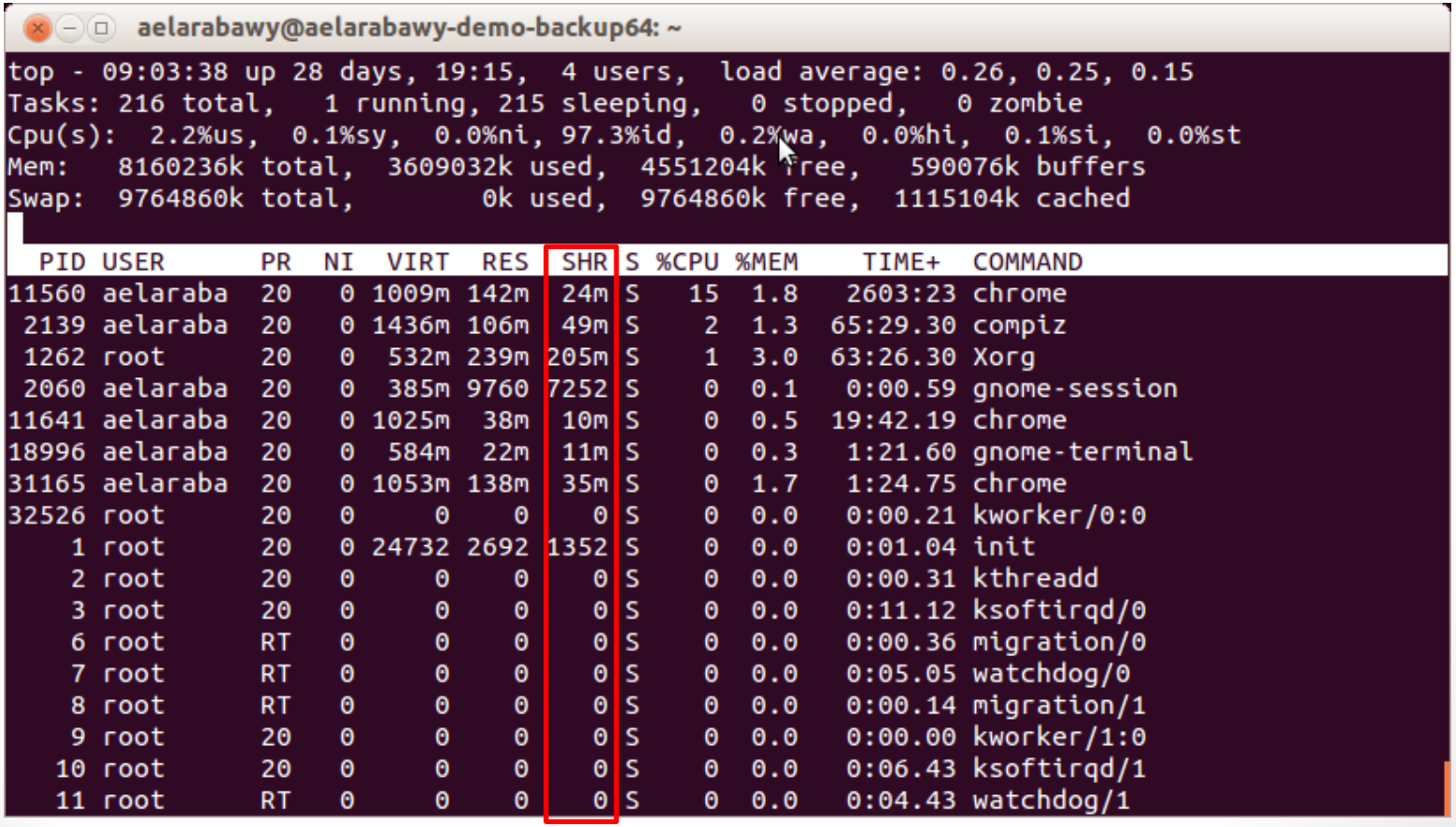
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

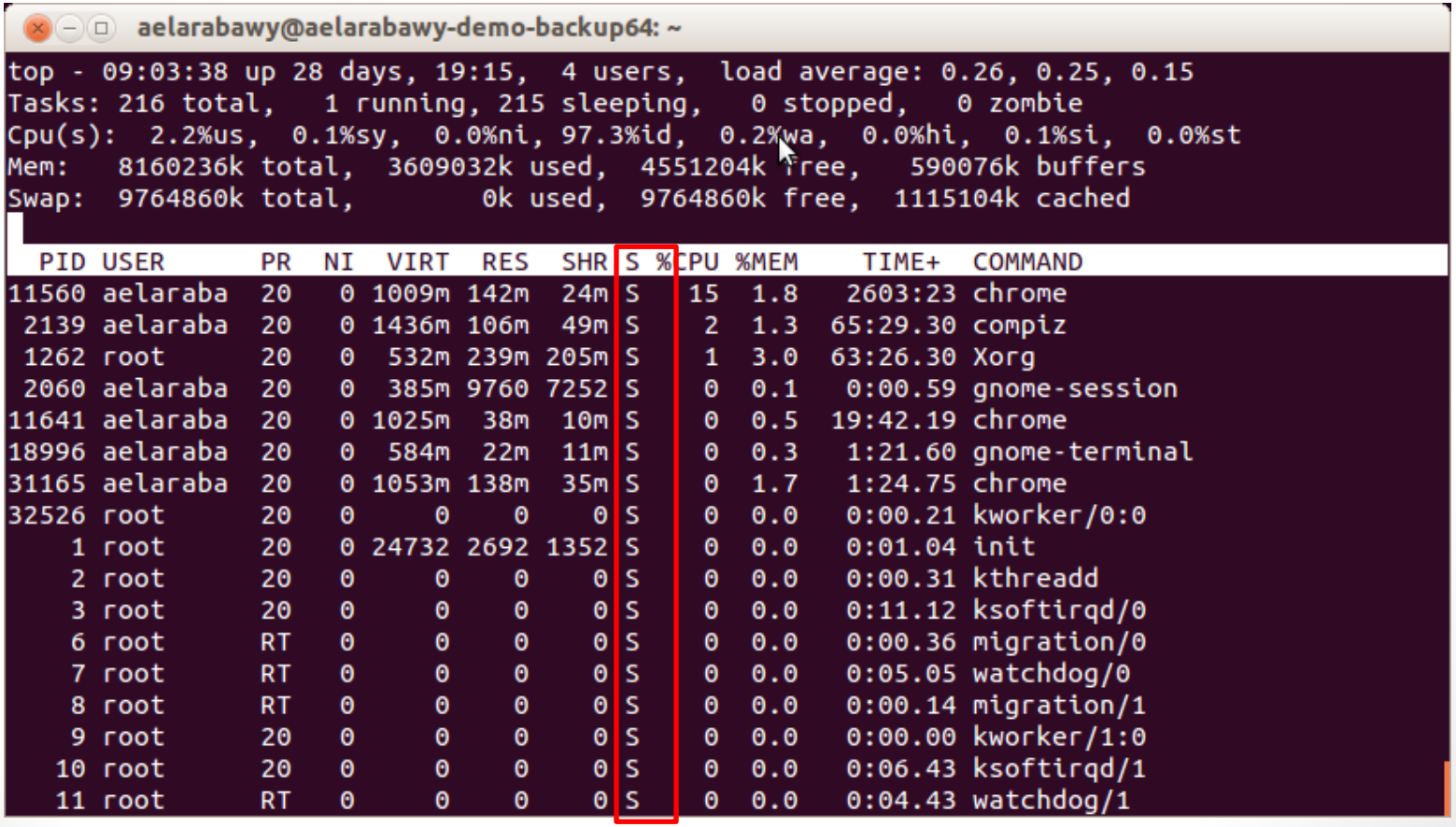
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

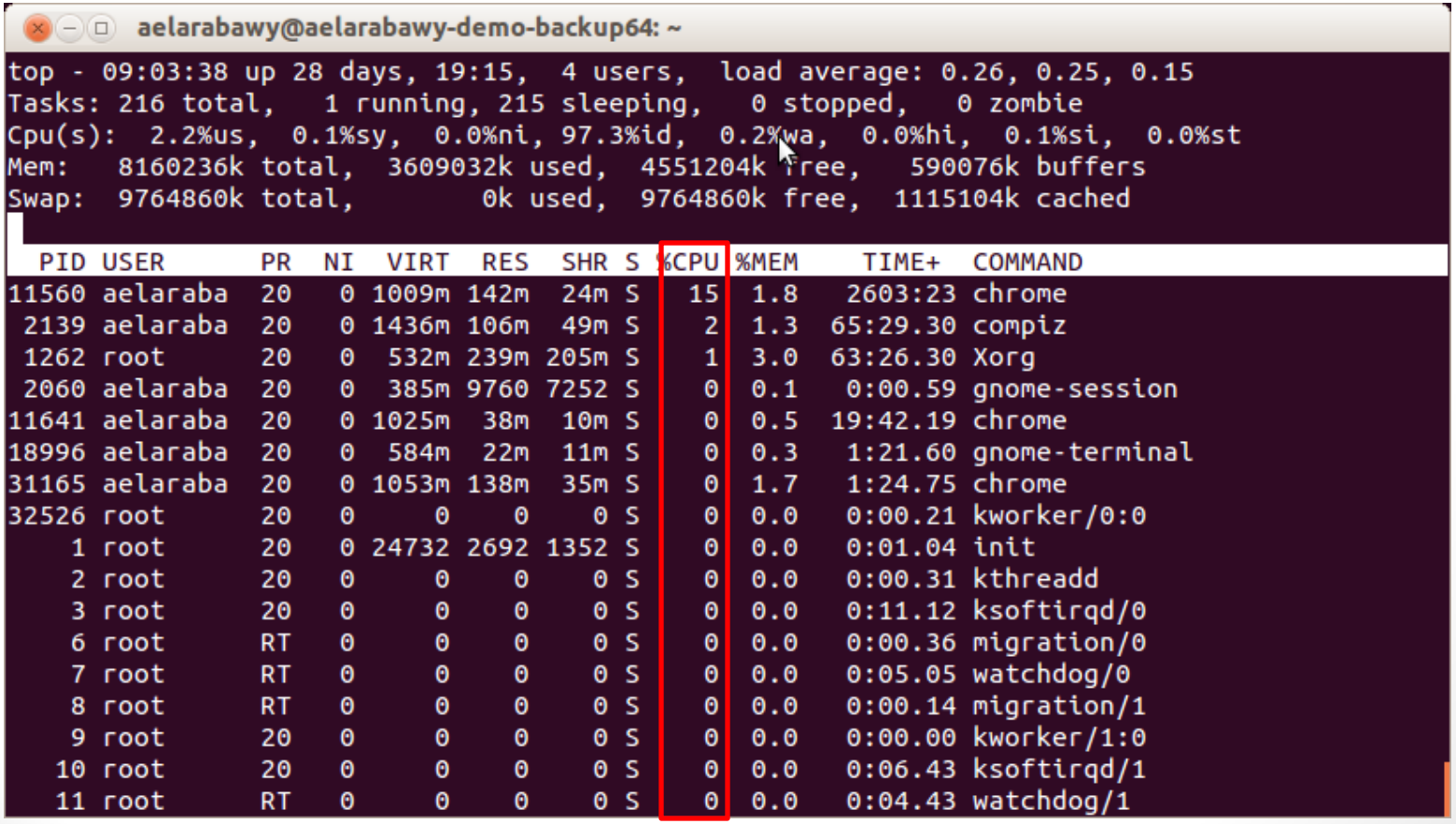
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

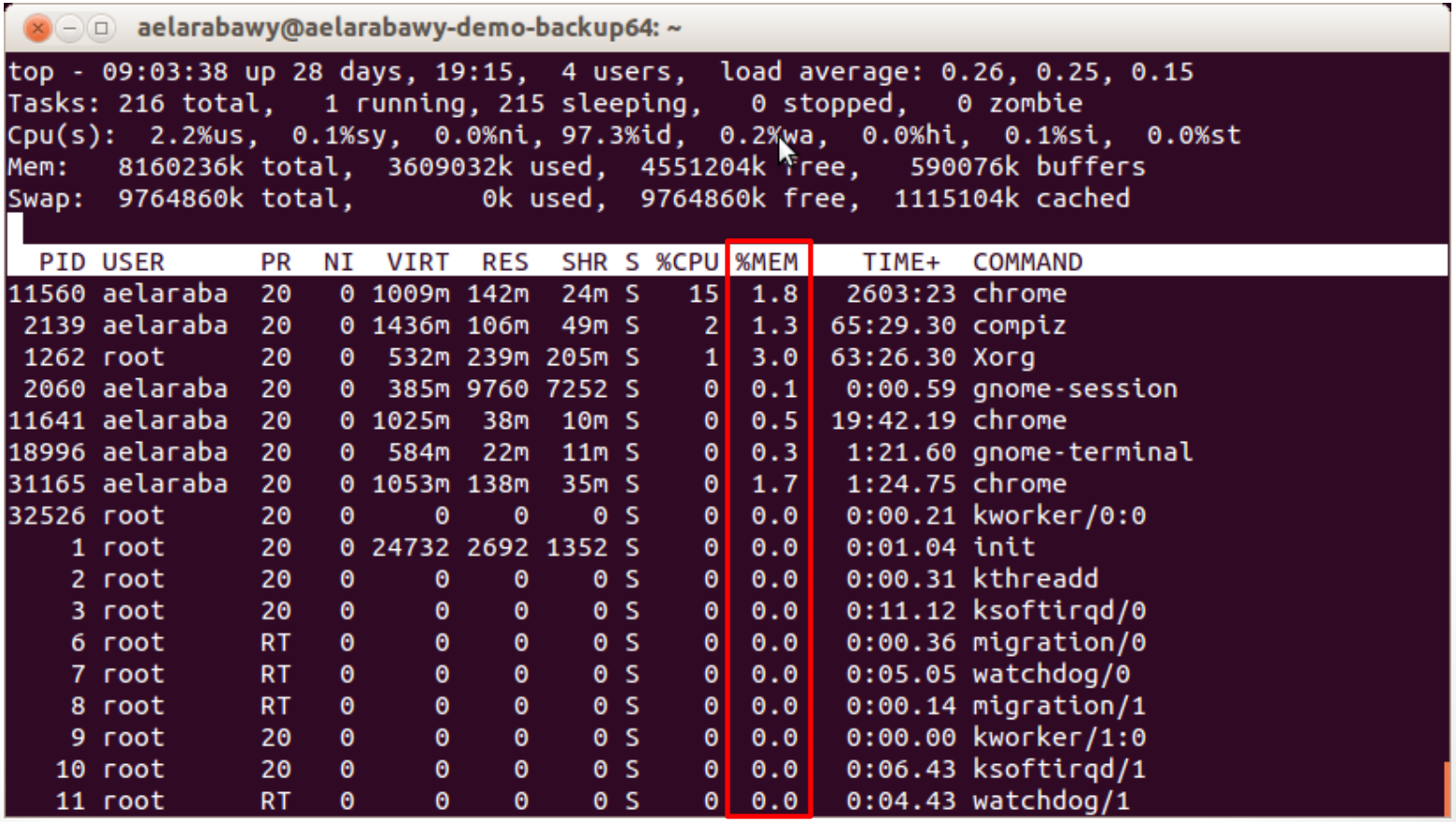
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

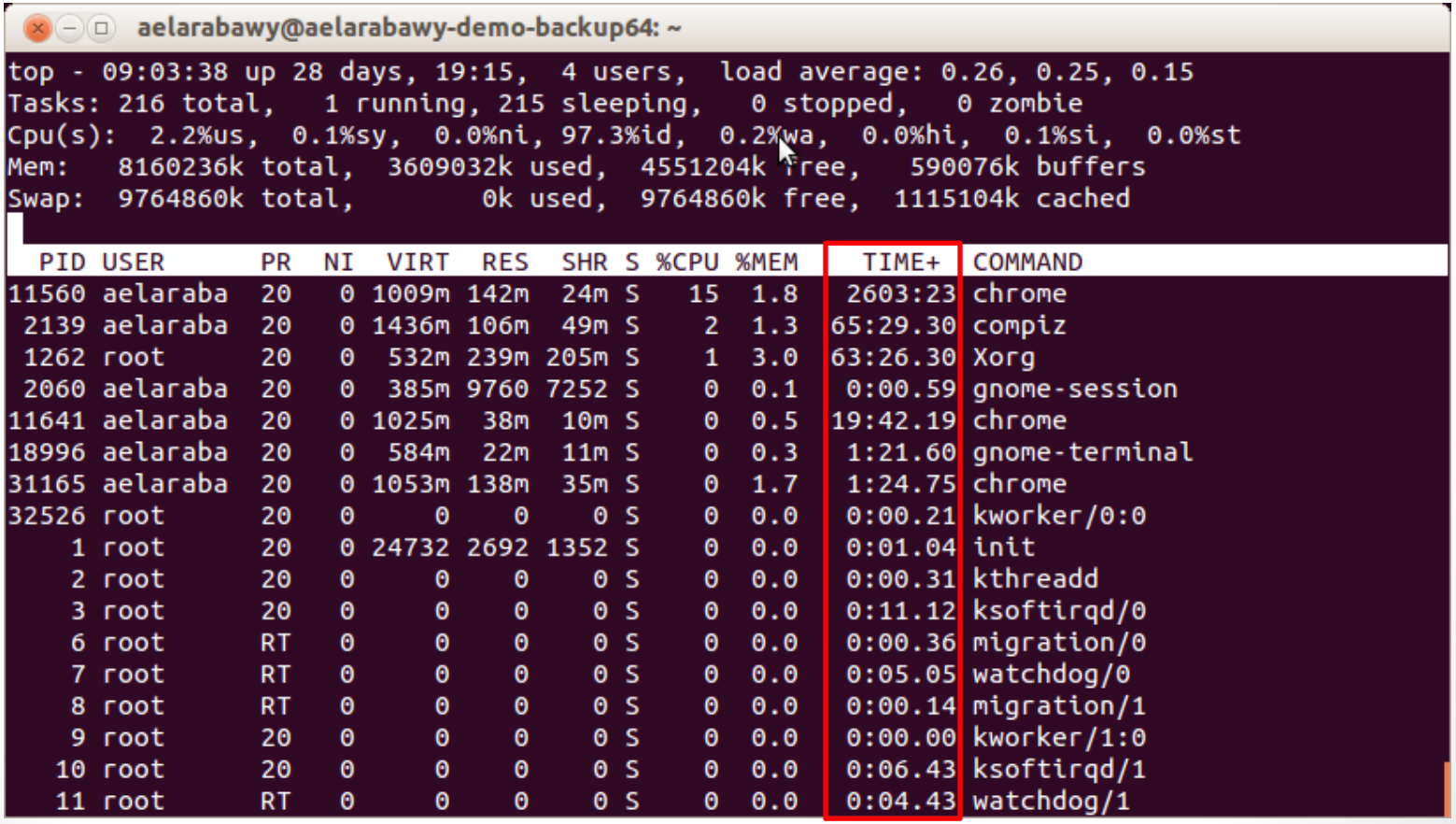
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

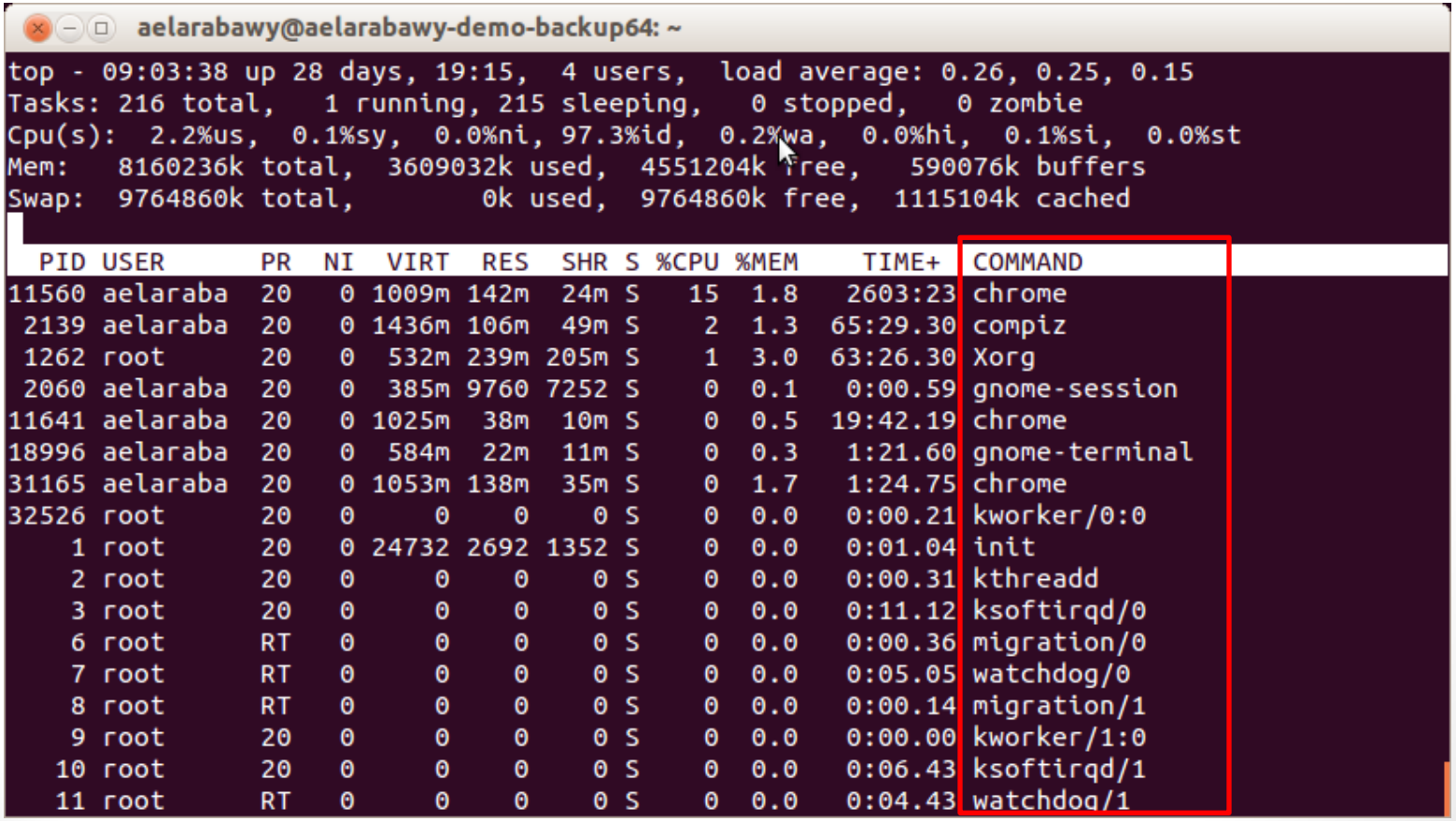
Displays a dynamic view of the resource usage of system processes



Display Process Resource Usage (top Command)

$ top

Displays a dynamic view of the resource usage of system processes



Manipulating Output

- While the "***top***" tool is running,
  - Push 'M' to sort by memory usage
  - Push 'P' to sort by CPU processing usage
  - Push 'T' to sort by Time
  - Push 'k <pid>' to kill process by its pid
  - Push 'h' for getting a help page for all options
  - Push 'H' to enable/disable showing threads separately
  - Push 'q' to quit the tool

Display System Resource Usage (vmstat Command)

$ vmstat

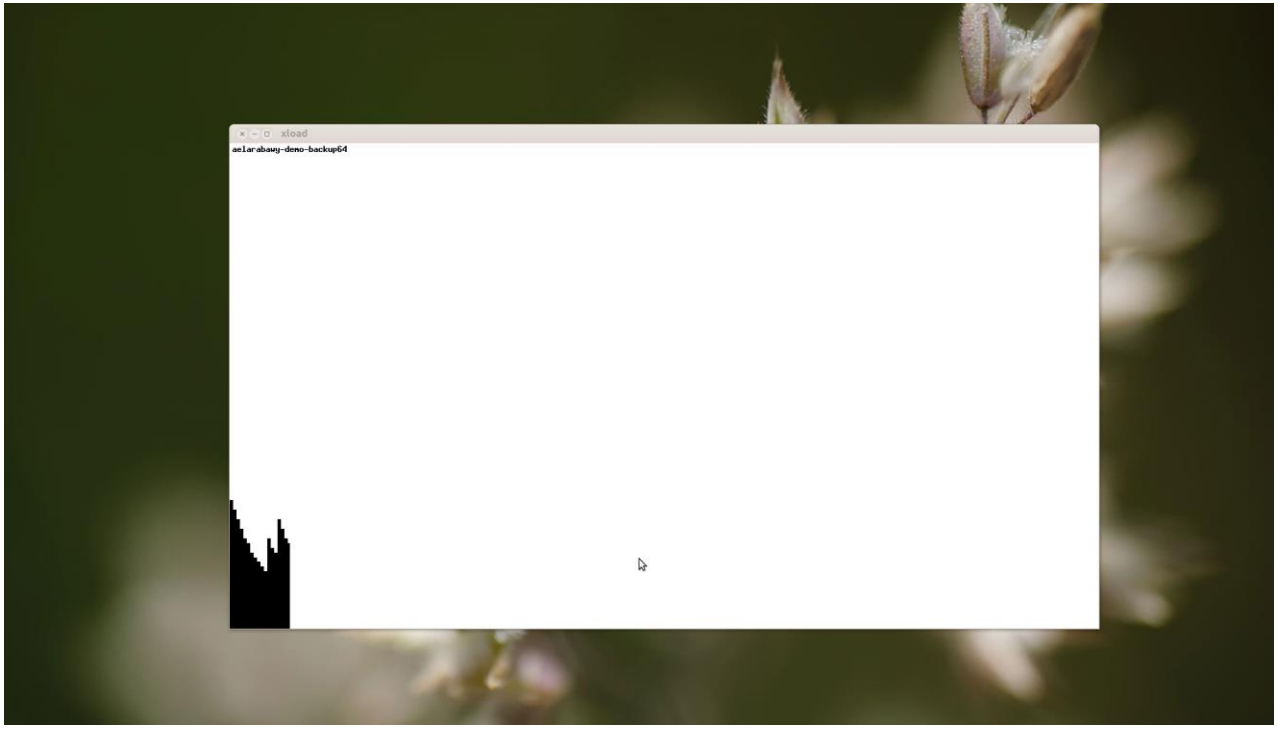
The ‘vmstat’ Command displays the system resource usage

```
root@dm816x:/opt/dvr_rdk/ti816x# vmstat 2
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
 0  1      0   2016      0 123100    0    0     0     0  838 1138  6  7 47 40
 1  1      0   2220      0 122656    0    0     0     0 6337  822  3  3  0 95
 1  1      0   2172      0 122688    0    0     0     0 6029  806  4  5  0 91
 1  1      0   2032      0 123160    0    0     0     0 7514  839  4  6  0 91
 0  1      0   2564      0 122288    0    0     0     0 4436  801  2  5  3 90
 0  0      0   3060      0 121836    0    0     0     0 1436  647  1  3 65 32
 0  0      0   2984      0 122004    0    0     0     0  513  622  1  2 97  0
 0  2      0   2092      0 122724    0    0     0     0 9130  680  1  5 35 58
 0  1      0   2484      0 122440    0    0     0     0 5161  673  2  4 11 85
 0  0      0   2920      0 121996    0    0     0     0 2580  668  4  2 23 71
 0  1      0   2208      0 122884    0    0     0     0 5390  661  2  4 10 84
 0  0      0   3108      0 121808    0    0     0     0 4965  671  2  3 14 82
 0  1      0   2184      0 122708    0    0     0     0 7554  652  4  3 16 78
 1  1      0   2084      0 122804    0    0     0     0 8197  683  3  4  5 89
 0  1      0   2280      0 122828    0    0     0     0 3173  641  2  3 23 73
 0  1      0   2204      0 122840    0    0     0     0 9421  663  3  5  0 92
 0  0      0   3208      0 121676    0    0     0     0 4501  642  0  3 20 76
 0  0      0   2984      0 122148    0    0     0     0  732  603  3  1 95  2
 0  1      0   2692      0 122164    0    0     0     0 5785  630  1  5 63 32
 0  0      0   3044      0 121816    0    0     0     0 5623  650  2  5 10 84
 0  1      0   1992      0 123052    0    0     0     0 6850  651  2  6  4 89
```

Display (xload Command)

$ xload

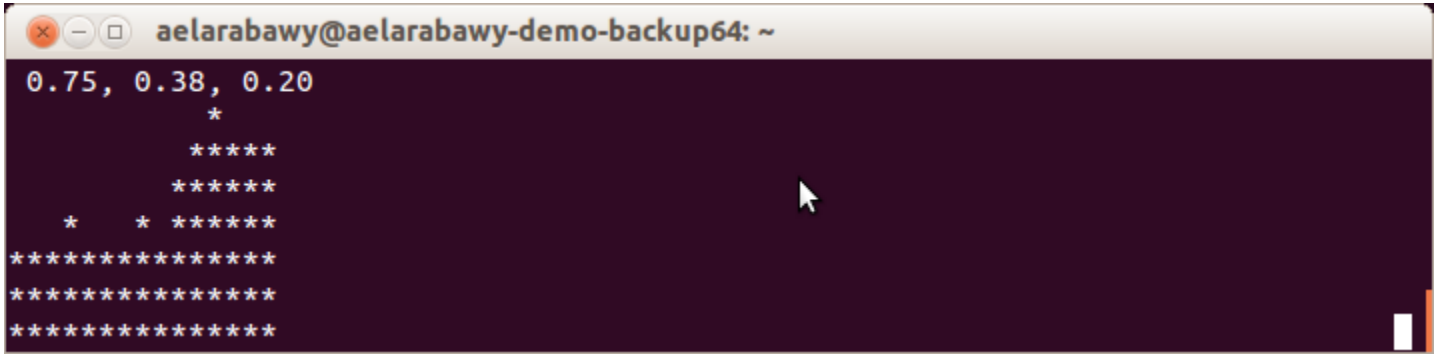
This command starts a graphical tool for measurement of system load average



Display (tload Command)

$ tload

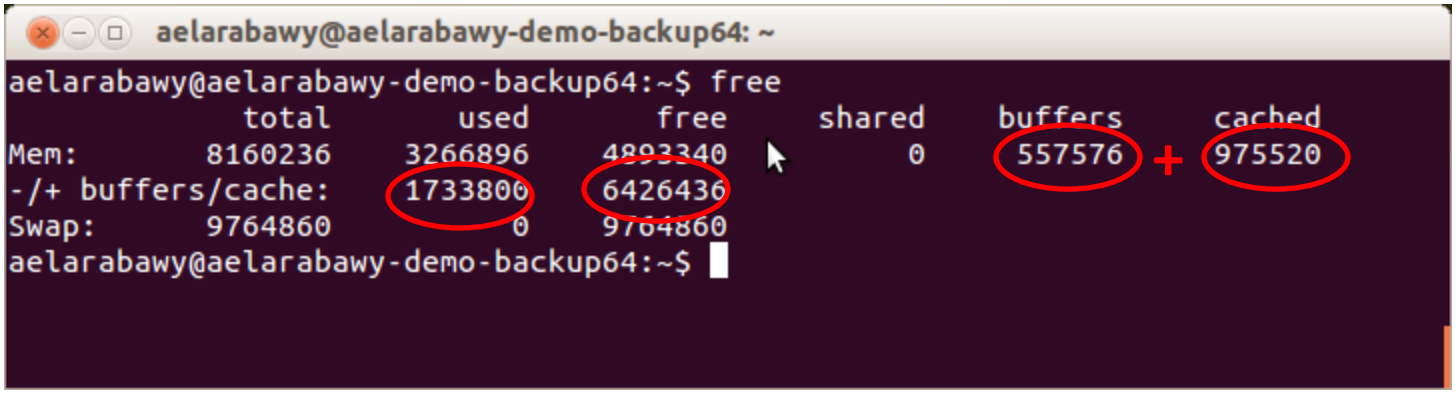
This command starts a graphical presentation (using characters) for measurement of system load average



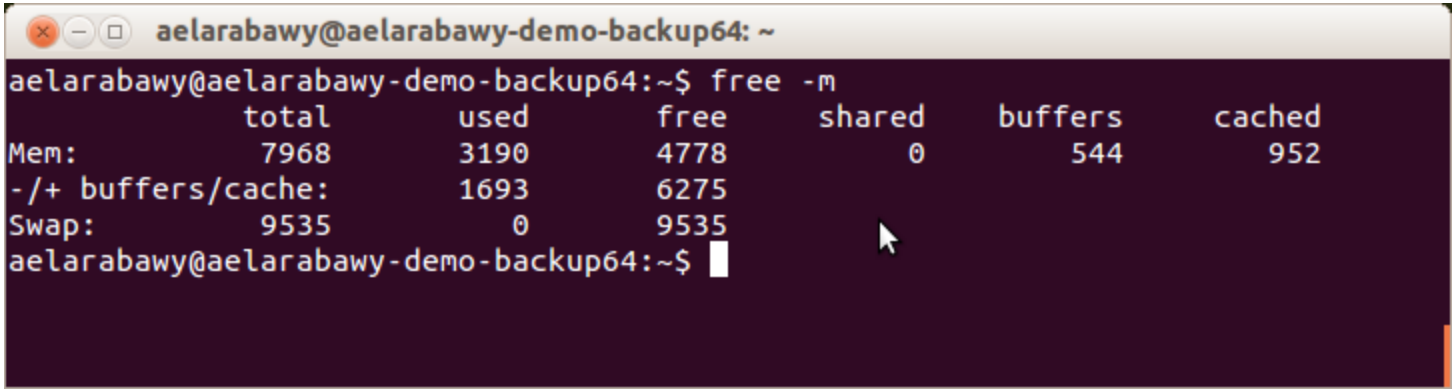
Display Memory Usage (free Command)

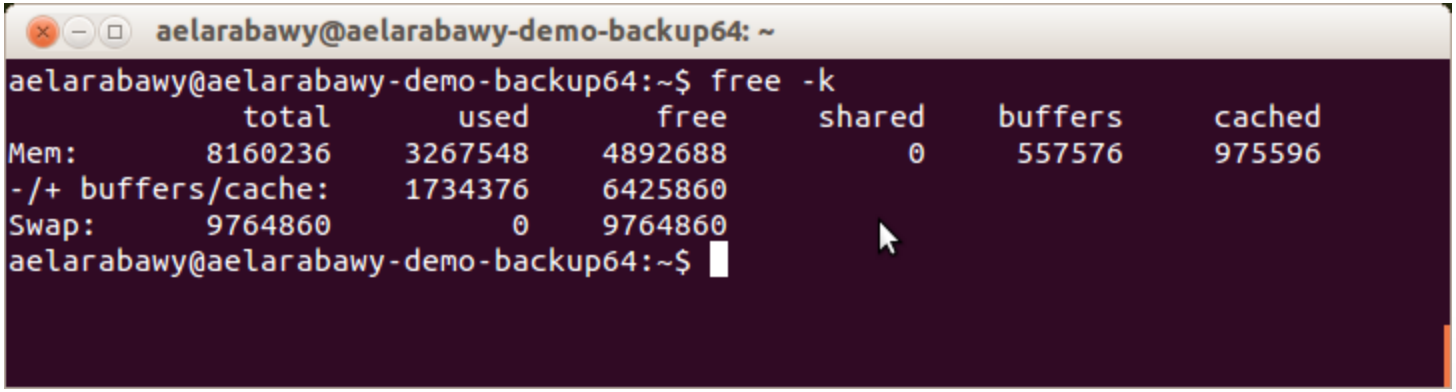
$ free <options>

This command displays amount of free and used memory in the system

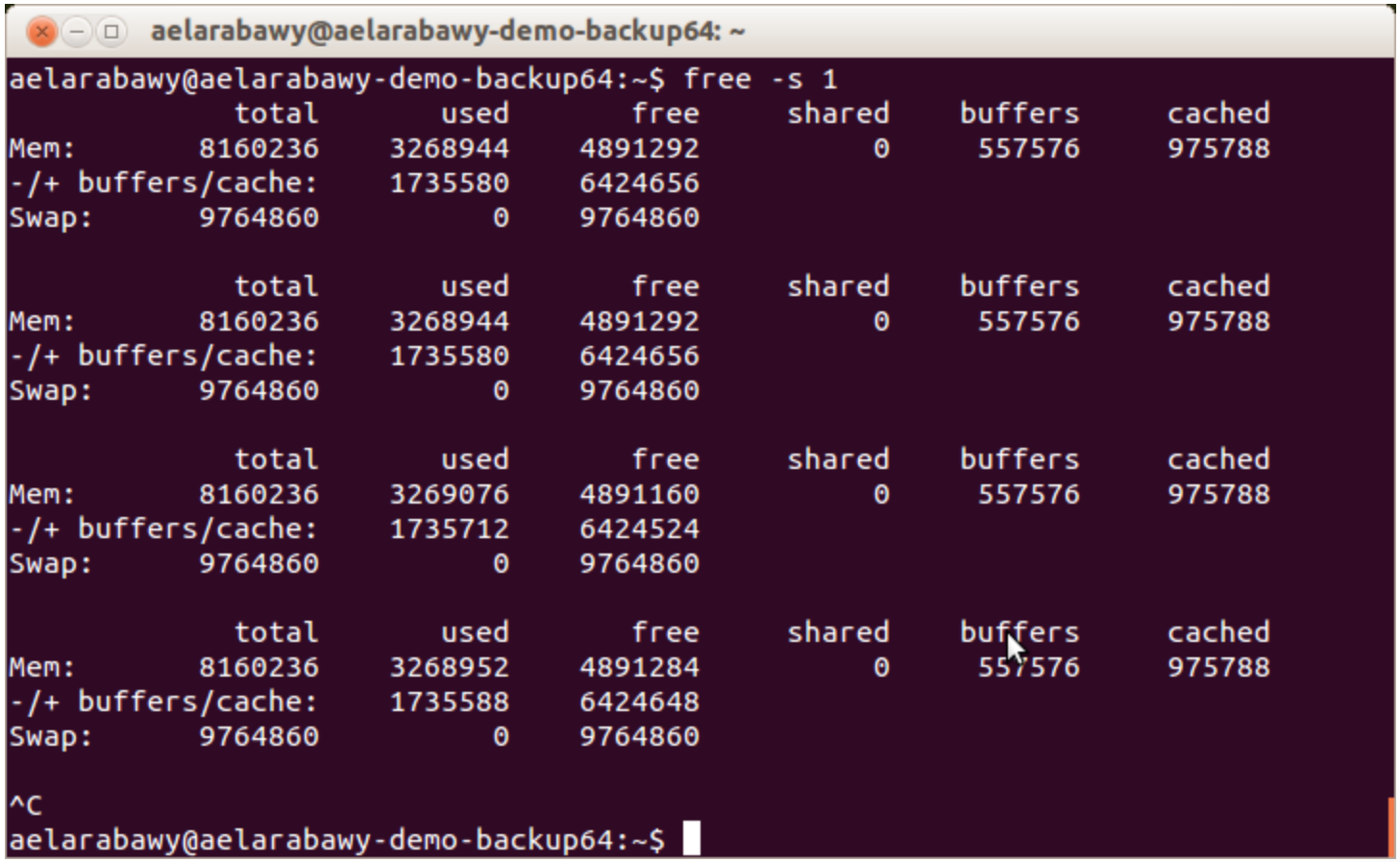


The **free** Command





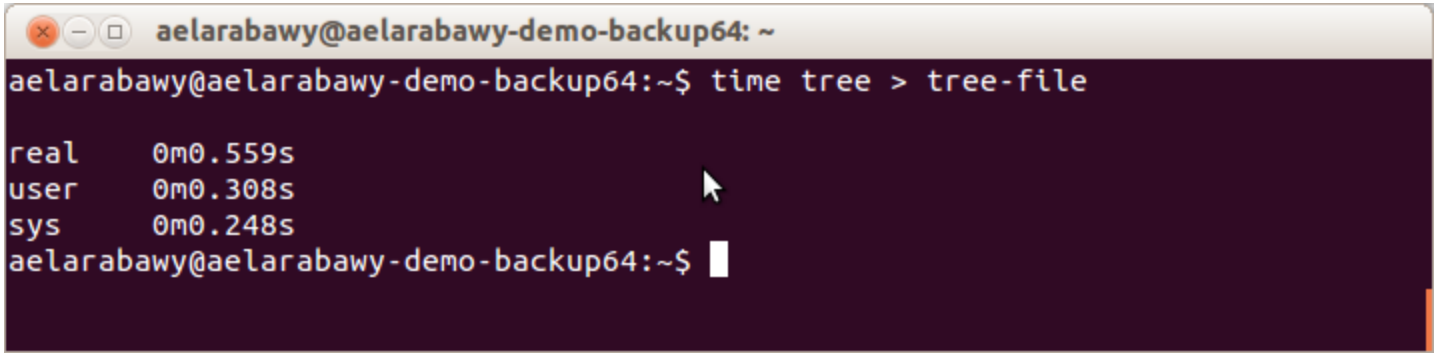
The **free** Command



Measuring Running Time (time Command)

$ time <command>

- Displays elapsed time running a command



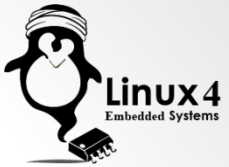
This will result in

- Output of the command
- Real time ➔ total time = user + system + waiting
- User time ➔ User space time
- System time ➔ system time

Note that the waiting time stands for the time the command spent waiting due to :

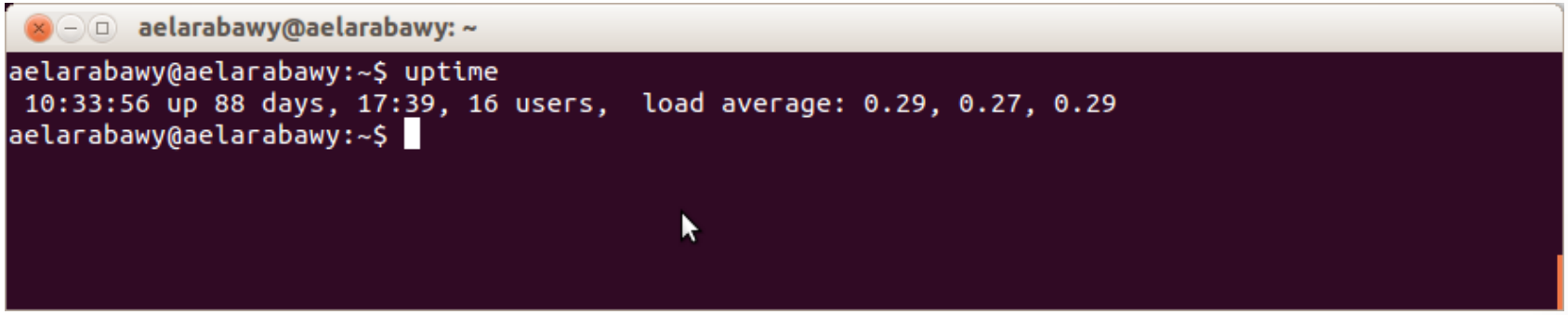
- I/O blocking
- Scheduling and multi-tasking (context switching)

Measuring System Uptime (uptime Command)



$ uptime

- Displays system up-time and load average



Load average,

- Calculated for the last 1, 5, and 15 minutes
- Note that average load is not normalized to number of system cores,

  For example, a load average of 1 means,
  - Fully loaded CPU in case of a single core processor
  - 25% loaded CPU in case of a Quad-core processor

Linux 4 Embedded Systems

http://Linux4EmbeddedSystems.com