Damir Nabiullin - Lab 7

1. Zombie process - process that completed execution but not is removed from the process table. Zombie processes usually mean an application or service didn't exit gracefully.

   We can find such processes using `ps aux | grep 'Z'`. This command returns table. We need to find rows where `STAT` column has `Z`.
   To kill process we need to get PID from `PID` column and use `kill -9 pid`.

2. kill - accepts PID and kills only process with this PID.
   killall - accepts a name of the process and kills all processes with the same name.
   pkill - look up or signal processes based on name and other attributes.

3. **Tasks:**
   1) `total`: I have 286 processes at all (in different states)
   2) `running`: 1 process is executting normally and have access to CPU
   3) `sleeping`: 285 processes are awaiting resources
   4) `stopped`: 0 processes are releasing resources and send termination message to parent process.
   5) `zombie`: o processes are zombie - process waiting for its parent process to release it.

   Cpu(s):
   1) `us`: 1% of time spent running user processes.
   2) `sy`: 0.8% of time spent running the kernel.
   3) `ni`: 0% of time spent running processes with manually configured nice values.
   4) `id`: 98.2 of time idle - CPU is overworked.
   5) `wa`: 0% of wait time.
   6) `hi`: 0% of time managing hardware interrupts.
   7) `si`: 0% of time managing software interrupts.
   8) `st`: 0% of virtual CPU time waiting for access to physical CPU.

```
top - 13:52:14 up  1:02,  1 user,  load average: 0.25, 0.60, 0.50
Tasks: 286 total,   1 running, 285 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.0 us,  0.8 sy,  0.0 ni, 98.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15837.0 total,  10522.2 free,   2288.8 used,   3026.0 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.  12578.4 avail Mem
```

4. I used such script:

```
#!/bin/bash
for file in $(ps -ef | grep 'fun[0-9][0-9]*process' | awk -F " " '{print $2}');
do
    $(kill -9 $file)
done
```

```
dale@dale-nitro:~/Documents/SNA/Week 7$ bash -c "exec -a fun${RANDOM}process sle
ep infinity" &
[1] 11986
dale@dale-nitro:~/Documents/SNA/Week 7$ bash -c "exec -a fun${RANDOM}process sle
ep infinity" &
[2] 11987
dale@dale-nitro:~/Documents/SNA/Week 7$ bash -c "exec -a fun${RANDOM}process sle
ep infinity" &
[3] 11988
dale@dale-nitro:~/Documents/SNA/Week 7$ bash task4.sh
[1]   Killed                  bash -c "exec -a fun${RANDOM}process sleep infinit
y"
[2]-  Killed                  bash -c "exec -a fun${RANDOM}process sleep infinit
y"
[3]+  Killed                  bash -c "exec -a fun${RANDOM}process sleep infinit
y"
dale@dale-nitro:~/Documents/SNA/Week 7$ ps -ef | grep 'fun[0-9][0-9]*process'
```

5. I used such script:

```
#!/bin/bash

trap "echo SIGUSR1 Interraupt received" SIGUSR1
while :
do
    echo "Hello, world"
    sleep 10
done
```

```
dale@dale-nitro:~/Documents/SNA/Week 7$ bash task5.sh & echo $!
[1] 11712
11712
dale@dale-nitro:~/Documents/SNA/Week 7$ Hello, world
SIGUSR1 Interraupt received
Hello, world
Hello, world
```

```
dale@dale-nitro:~$ kill -SIGUSR1 11712
dale@dale-nitro:~$
```

6. I used such script:

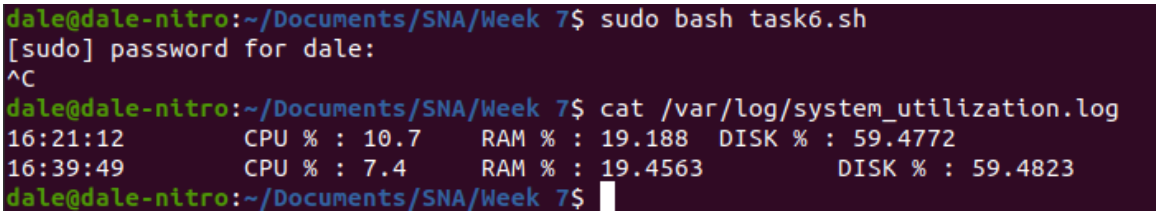
```
#!/bin/bash

CURDISK=$(awk '$2 == "/" {print $1}' /proc/self/mounts)
logdir=$"/var/log/system_utilization.log"

while :
do
    TIME_N=$(top -bn 2 -d 0.01 | grep '^top -' | head -n 1 | awk '{print $3}')
    CPU_U=$(top -bn 2 -d 0.01 | grep '^%Cpu' | tail -n 1 | awk '{print 100-$8}')
    RAM_U=$(top -bn 2 -d 0.01 | grep '^MiB Mem' | tail -n 1 | awk '{print ($8/$4)*100}')
    DISK_U=$(df | grep $CURDISK | awk '{print ($3/$2)*100}')

    echo -e $TIME_N "\tCPU % :" $CPU_U "\tRAM % :" $RAM_U "\tDISK % :" $DISK_U >> $logdir

    sleep 15
done
```


dale@dale-nitro:~/Documents/SNA/Week 7$ sudo bash task6.sh
[sudo] password for dale:
^C
dale@dale-nitro:~/Documents/SNA/Week 7$ cat /var/log/system_utilization.log
16:21:12 CPU % : 10.7 RAM % : 19.188 DISK % : 59.4772
16:39:49 CPU % : 7.4 RAM % : 19.4563 DISK % : 59.4823
dale@dale-nitro:~/Documents/SNA/Week 7$