LAB 3
LINUX PROGRAMMING

19-12-2019

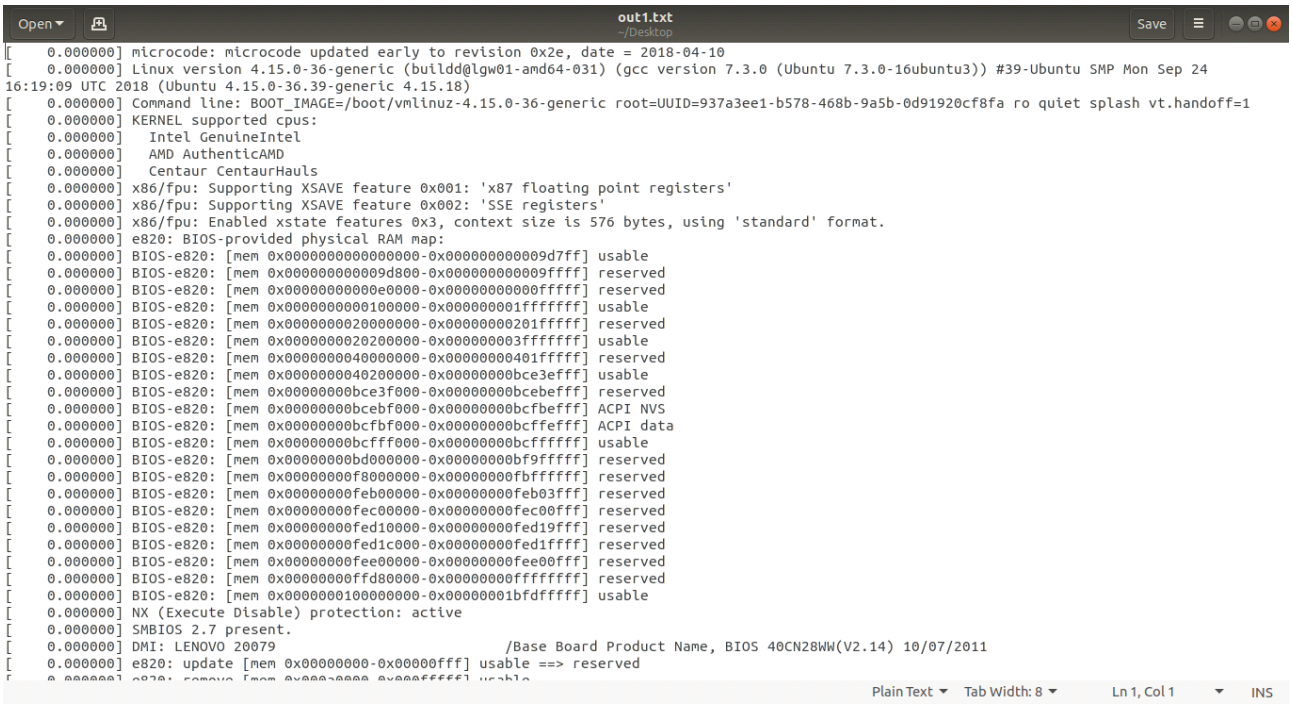
17MIS1086
K.prudhvi

1. using “grep” command

```
Lenovo-G570:~$ cd Desktop
Lenovo-G570:~/Desktop$ gedit try1.sh &

Lenovo-G570:~/Desktop$ ./try1.sh
.sh: Permission denied
Lenovo-G570:~/Desktop$ chmod +x try1.sh
Lenovo-G570:~/Desktop$ ./try1.sh
Lenovo-G570:~/Desktop$ dmesg > out1.txt
Lenovo-G570:~/Desktop$ grep-w "main" out1.txt
and not found
Lenovo-G570:~/Desktop$ grep -w "main" out1.txt
Lenovo-G570:~/Desktop$
```

Output:



2. Using grep in a while loop

try2.sh

```
#!/bin/sh
input="out1.txt"
while IFS= read -r line
do
        grep -w "error" > out2.txt
done< "$input"
```

File Edit View Search Terminal Help

```
Lenovo-G570:~$ cd Desktop
Lenovo-G570:~/Desktop$ chmod +x try2.sh
Lenovo-G570:~/Desktop$ ./try2.sh
Lenovo-G570:~/Desktop$ cat out2.txt
4] rtc_cmos: probe of 00:01 failed with error -16
Lenovo-G570:~/Desktop$
```

3. Automatically checking for errors every hour

```
Lenovo-G570:~/Desktop$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
```

Code:

```
#!/bin/sh
input="out1.txt"
while IFS= read -r line
do
        grep -w "error" > out3.txt
done< "$input"
```

Output:

```
Lenovo-G570:~/Desktop$ ls
certificate.pem   out2.txt                                        try2.sh
lab3.odt          'Screenshot from 2019-07-31 16-34-58.png'       try3.sh
out1.txt          'Screenshot from 2019-07-31 16-36-11.png'
```

```
Lenovo-G570:~/Desktop$ ls
certificate.pem
lab3.odt
out1.txt
out2.txt
out3.txt
'Screenshot from 2019-07-31 16-34-58.png'
'Screenshot from 2019-07-31 16-36-11.png'
'Screenshot from 2019-12-23 15-43-26.png'
'Screenshot from 2019-12-23 15-44-44.png'
'Screenshot from 2019-12-23 15-45-43.png'
try2.sh
try3.sh
```

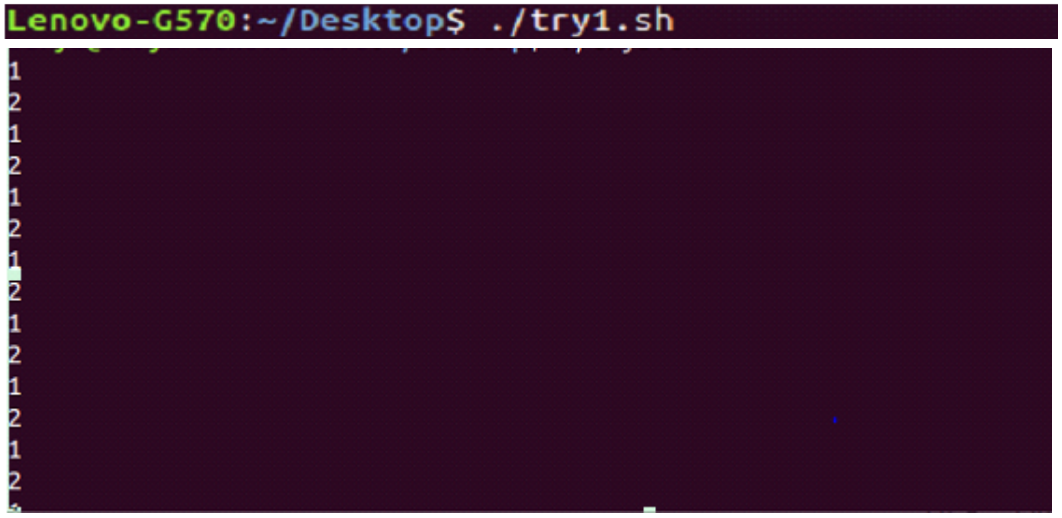
4.Monitoring memory performance of a process (running background) for every 5 minutes...

let’s say the the process in “try5.sh”:

try5.sh

```
#!/bin/sh
v=1
while [ $v -le 10000 ]
do
        echo $v
        v= expr $v + 1
        sleep 2
done
```

OUTPUT OF try5.sh:

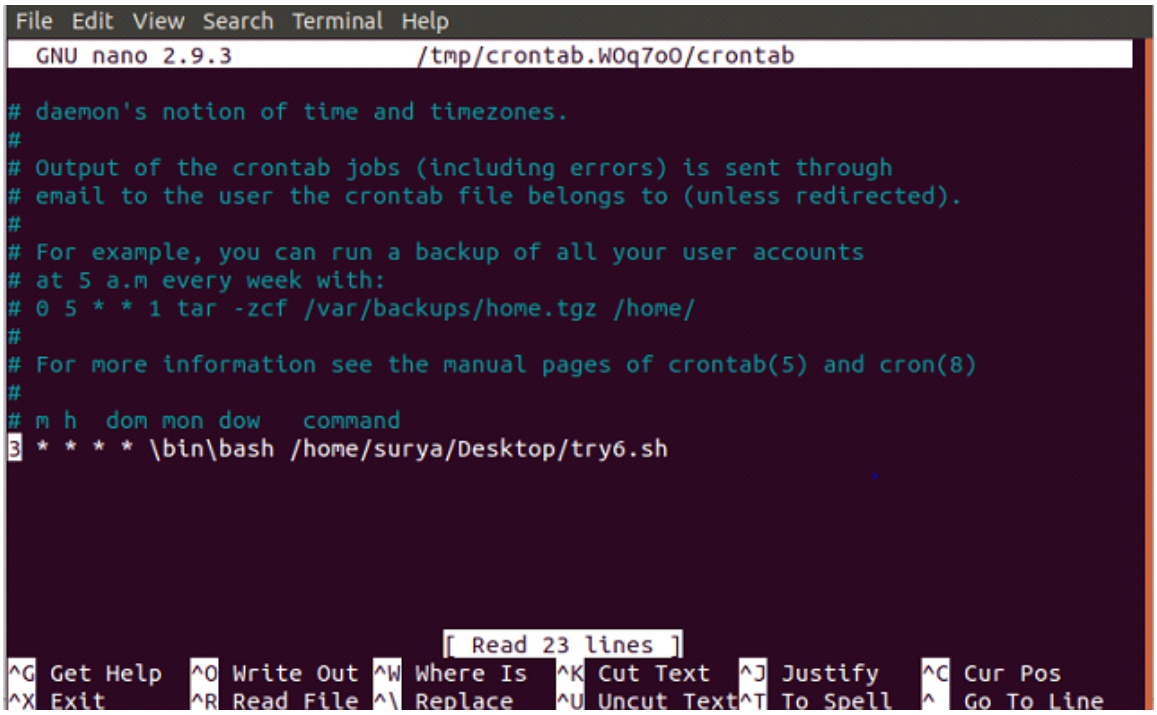


So the process id of try.5.sh is 29021 , now let’s create another process try6.sh and get the memory performance details for that id using top command....

try6.sh:

So we need to automate the excecution of try6.sh. This can be done by crontab.

creating a crontab....




```
Lenovo-G570:~/Desktop$ crontab -e
crontab: installing new crontab
```

```
Lenovo-G570:~/Desktop$ ls
certificate.pem   'Screenshot from 2019-07-31 16-34-58.png'   try2.sh
lab3.odt          'Screenshot from 2019-07-31 16-36-11.png'   try3.sh
out1.txt          'Screenshot from 2019-12-23 16-13-44.png'   try5.sh
out2.txt          'Screenshot from 2019-12-23 16-14-11.png'   try6.sh
out3.txt          'Screenshot from 2019-12-23 16-15-45.png'
```

After 3 mins:

```
Lenovo-G570:~/Desktop$ ls
```

```
certificate.pem
lab3.odt
out1.txt
out2.txt
out3.txt
'Screenshot from 2019-07-31 16-34-58.png'
'Screenshot from 2019-07-31 16-36-11.png'
'Screenshot from 2019-12-23 16-13-44.png'
'Screenshot from 2019-12-23 16-14-11.png'
'Screenshot from 2019-12-23 16-15-45.png'
'Screenshot from 2019-12-23 16-17-25.png'
try2.sh
try3.sh
try5.sh
try6.sh
```

```
Lenovo-G570:~/Desktop$ cat n8.txt
top - 16:24:05 up  1:56,  1 user,  load average: 0.39, 0.41, 0.37
Tasks:   0 total,   0 running,   0 sleeping,   0 stopped,   0 zombie
%Cpu(s):  7.8 us,  1.3 sy,  0.2 ni, 88.7 id,  2.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  6045772 total,  2895536 free,  1391076 used,  1759160 buff/cache
KiB Swap:  3952636 total,  3952636 free,        0 used.  4011948 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND

top - 16:24:08 up  1:56,  1 user,  load average: 0.36, 0.40, 0.37
Tasks:   0 total,   0 running,   0 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.2 us,  0.8 sy,  0.0 ni, 95.5 id,  1.3 wa,  0.0 hi,  0.2 si,  0.0 st
KiB Mem :  6045772 total,  2907068 free,  1391528 used,  1747176 buff/cache
KiB Swap:  3952636 total,  3952636 free,        0 used.  4023516 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND

top - 16:24:11 up  1:56,  1 user,  load average: 0.36, 0.40, 0.37
Tasks:   0 total,   0 running,   0 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.0 us,  0.3 sy,  0.0 ni, 97.8 id,  0.8 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  6045772 total,  2907036 free,  1391680 used,  1747056 buff/cache
KiB Swap:  3952636 total,  3952636 free,        0 used.  4023488 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
```