Linux Programming

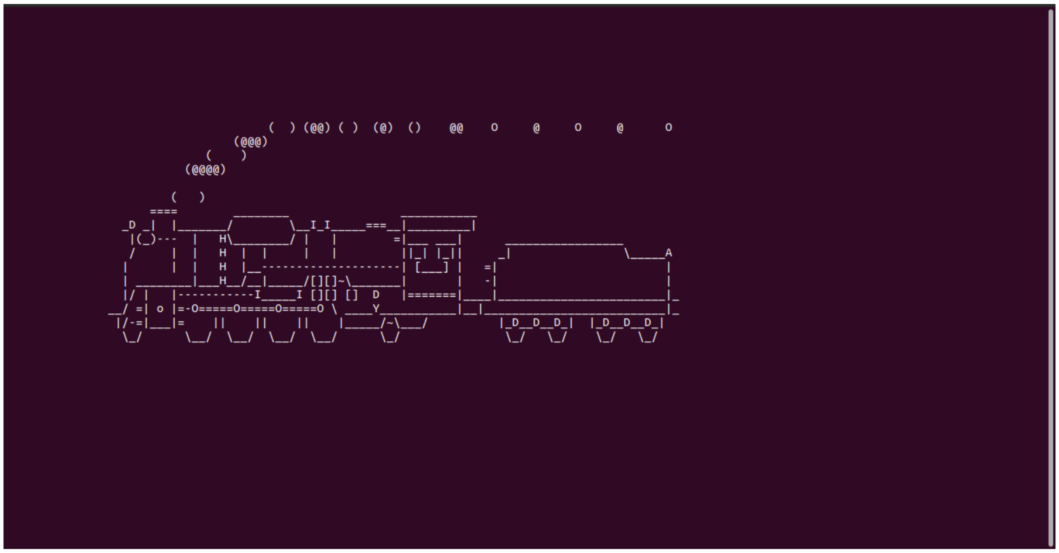
Lab Exercise -June

Prof.Harini S

Karthik Kurella
17MIS1022

1)sl –*funny train runs in terminal:*

Screenshots:



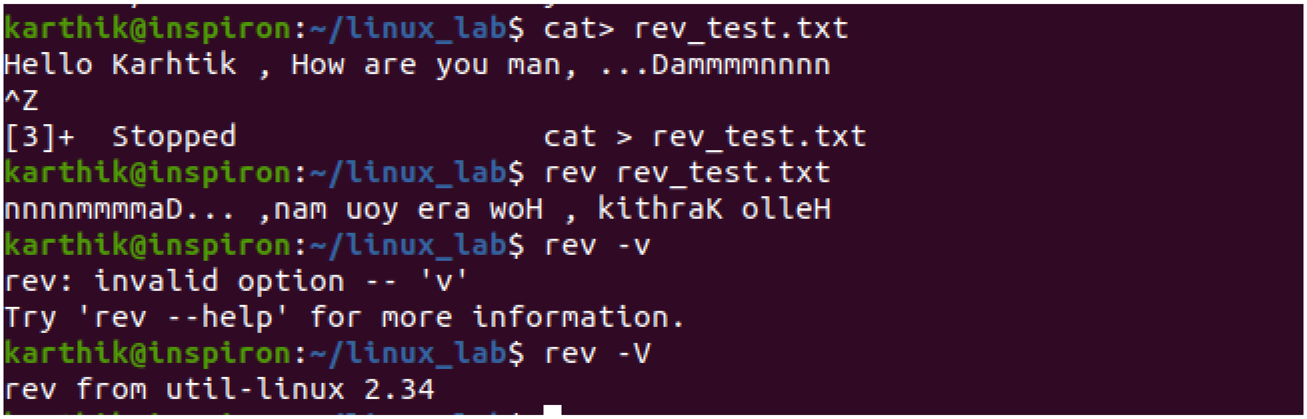
```
                        (  ) (@@) ( )  (@)  ()    @@     O     @     O     @      O
                   (@@@)
              (     )
           (@@@@)

         (   )
      ====        ________                ___________
  _D _|  |_______/        \__I_I_____===__|_________|
   |(_)---  |   H\________/ |   |        =|___ ___|      _________________
   /     |  |   H  |  |     |   |         ||_| |_||     _|                \_____A
  |      |  |   H  |__--------------------| [___] |   =|                        |
  | ________|___H__/__|_____/[][]~\_______|       |   -|                        |
  |/ |   |-----------I_____I [][] []  D   |=======|____|________________________|_
__/ =| o |=-O=====O=====O=====O \ ____Y___________|__|__________________________|_
 |/-=|___|=    ||    ||    ||    |_____/~\___/          |_D__D__D_|  |_D__D__D_|
  \_/      \__/  \__/  \__/  \__/      \_/               \_/   \_/    \_/   \_/
```

2) Rev Command in Linux:

Usage: *Rev [text] or[filename]*

```
rev -h ---Help
rev -V ---Version Number
```

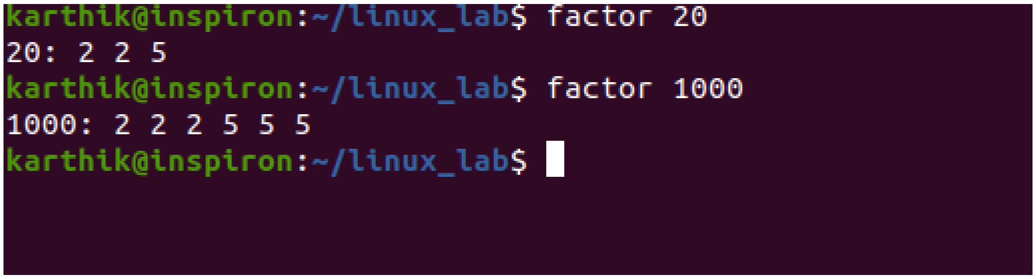
Screenshots:



```
karthik@inspiron:~/linux_lab$ cat> rev_test.txt
Hello Karhtik , How are you man, ...Dammmmnnnn
^Z
[3]+  Stopped                 cat > rev_test.txt
karthik@inspiron:~/linux_lab$ rev rev_test.txt
nnnnmmmmaD... ,nam uoy era woH , kithraK olleH
karthik@inspiron:~/linux_lab$ rev -v
rev: invalid option -- 'v'
Try 'rev --help' for more information.
karthik@inspiron:~/linux_lab$ rev -V
rev from util-linux 2.34
```

3)**Factor:***The factor command in Linux is used to print the prime factors of the given numbers.*

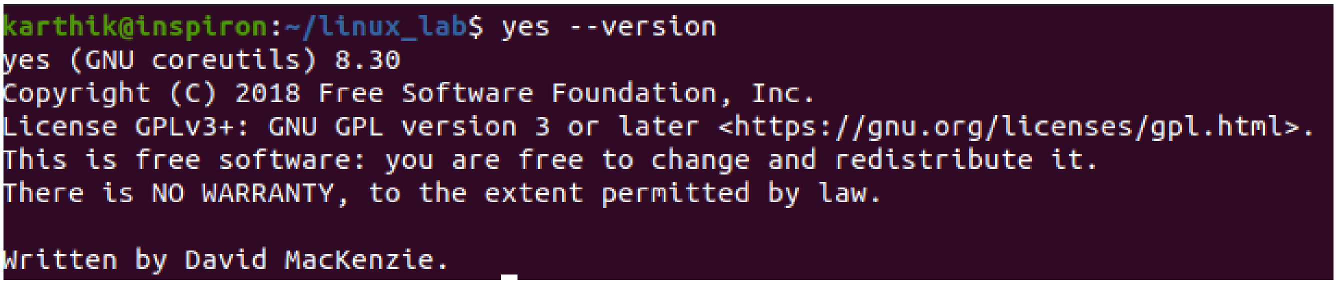
Screenshots:



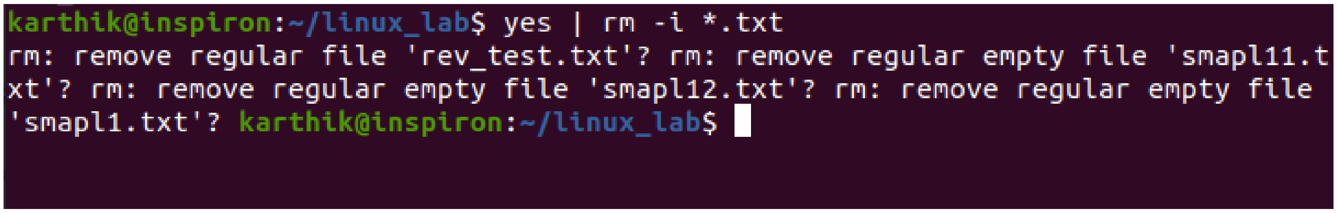
```
karthik@inspiron:~/linux_lab$ factor 20
20: 2 2 5
karthik@inspiron:~/linux_lab$ factor 1000
1000: 2 2 2 5 5 5
karthik@inspiron:~/linux_lab$
```

*4)**yes:** y*es command in linux is used to print a continuous output stream of given *STRING*. If *STRING* is not mentioned then it prints ‘y’

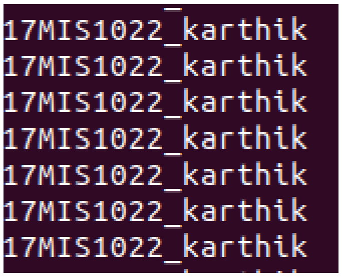
Screenshots:



```
karthik@inspiron:~/linux_lab$ yes --version
yes (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie.
```



```
karthik@inspiron:~/linux_lab$ yes | rm -i *.txt
rm: remove regular file 'rev_test.txt'? rm: remove regular empty file 'smapl11.t
xt'? rm: remove regular empty file 'smapl12.txt'? rm: remove regular empty file
'smapl1.txt'? karthik@inspiron:~/linux_lab$ 
```



```
17MIS1022_karthik
17MIS1022_karthik
17MIS1022_karthik
17MIS1022_karthik
17MIS1022_karthik
17MIS1022_karthik
17MIS1022_karthik
```

Write a bash shell script to monitor the health of your system. Let the details be stored and archived in any folder of your choice.

Instructions:

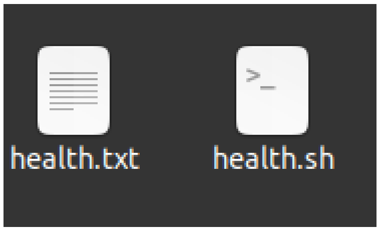
crontab -e --- to install the shell script for automation

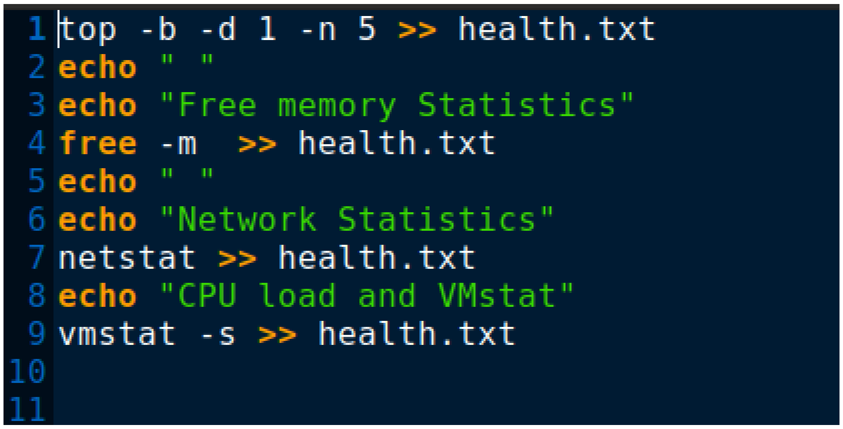
Health monitor used in the scenario:

Top---process info
free---memory usage in the system
netstat---network info and socket connected info
vmstat—virtual mem and cache info

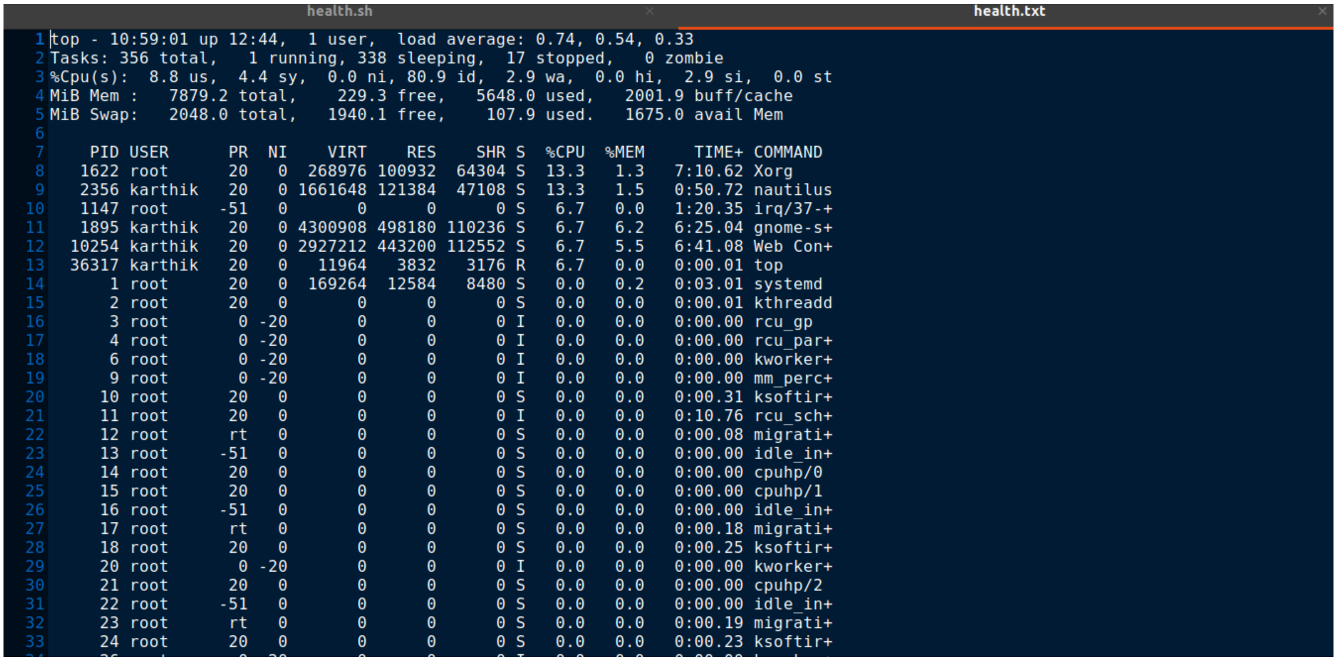
Screenshots:

files: health.sh and health.txt



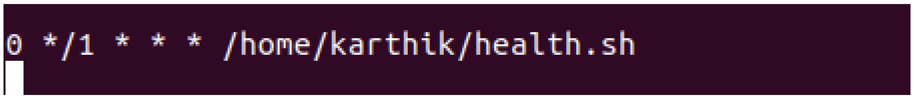


```
top -b -d 1 -n 5 >> health.txt
echo " "
echo "Free memory Statistics"
free -m  >> health.txt
echo " "
echo "Network Statistics"
netstat >> health.txt
echo "CPU load and VMstat"
vmstat -s >> health.txt


```



```
top - 10:59:01 up 12:44,  1 user,  load average: 0.74, 0.54, 0.33
Tasks: 356 total,   1 running, 338 sleeping,  17 stopped,   0 zombie
%Cpu(s):  8.8 us,  4.4 sy,  0.0 ni, 80.9 id,  2.9 wa,  0.0 hi,  2.9 si,  0.0 st
MiB Mem :   7879.2 total,    229.3 free,   5648.0 used,   2001.9 buff/cache
MiB Swap:   2048.0 total,   1940.1 free,    107.9 used.   1675.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1622 root      20   0  268976 100932  64304 S  13.3   1.3   7:10.62 Xorg
   2356 karthik   20   0 1661648 121384  47108 S  13.3   1.5   0:50.72 nautilus
   1147 root     -51   0       0      0      0 S   6.7   0.0   1:20.35 irq/37-+
   1895 karthik   20   0 4300908 498180 110236 S   6.7   6.2   6:25.04 gnome-s+
  10254 karthik   20   0 2927212 443200 112552 S   6.7   5.5   6:41.08 Web Con+
  36317 karthik   20   0   11964   3832   3176 R   6.7   0.0   0:00.01 top
      1 root      20   0  169264  12584   8480 S   0.0   0.2   0:03.01 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 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+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.31 ksoftir+
     11 root      20   0       0      0      0 I   0.0   0.0   0:10.76 rcu_sch+
     12 root      rt   0       0      0      0 S   0.0   0.0   0:00.08 migrati+
     13 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_in+
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_in+
     17 root      rt   0       0      0      0 S   0.0   0.0   0:00.18 migrati+
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.25 ksoftir+
     20 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     21 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
     22 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_in+
     23 root      rt   0       0      0      0 S   0.0   0.0   0:00.19 migrati+
     24 root      20   0       0      0      0 S   0.0   0.0   0:00.23 ksoftir+
```

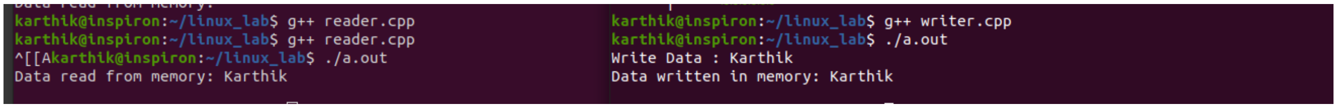
Crontab :



```
0 */1 * * * /home/karthik/health.sh
```

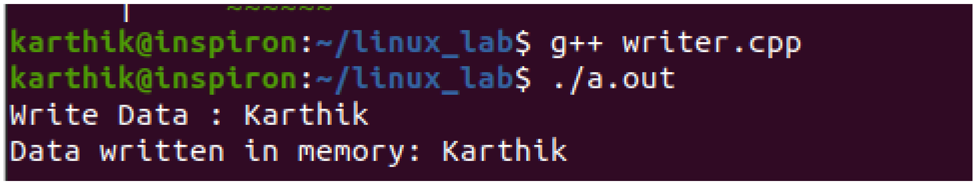
Write a C program to implement Simple reader – writer algorithm using shared memory segment with semaphore

A)

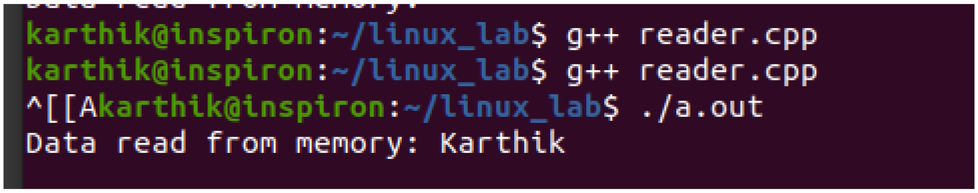


```
Data read from memory:
karthik@inspiron:~/linux_lab$ g++ reader.cpp
karthik@inspiron:~/linux_lab$ g++ reader.cpp
^[[Akarthik@inspiron:~/linux_lab$ ./a.out
Data read from memory: Karthik
```

```
karthik@inspiron:~/linux_lab$ g++ writer.cpp
karthik@inspiron:~/linux_lab$ ./a.out
Write Data : Karthik
Data written in memory: Karthik
```



```
karthik@inspiron:~/linux_lab$ g++ writer.cpp
karthik@inspiron:~/linux_lab$ ./a.out
Write Data : Karthik
Data written in memory: Karthik
```



```
Data read from memory:
karthik@inspiron:~/linux_lab$ g++ reader.cpp
karthik@inspiron:~/linux_lab$ g++ reader.cpp
^[[Akarthik@inspiron:~/linux_lab$ ./a.out
Data read from memory: Karthik
```