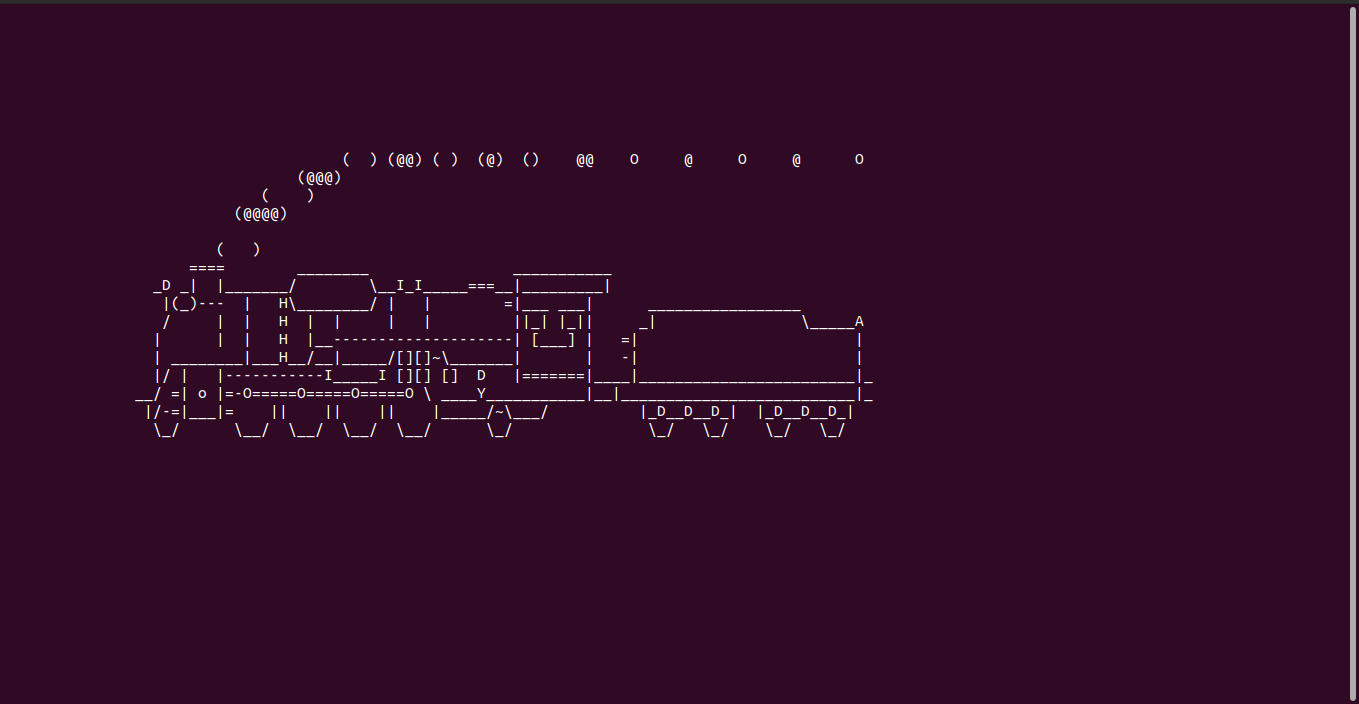
**17MIS1139**

**S.SOWMIYA**

**LAB EXERCISE JUNE**

1. **Script1**

**Sl –** funny train runs in terminal

Output:

**Hot Question:**

**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:

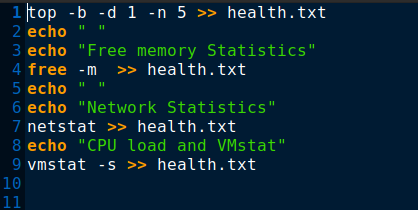
Top---process info

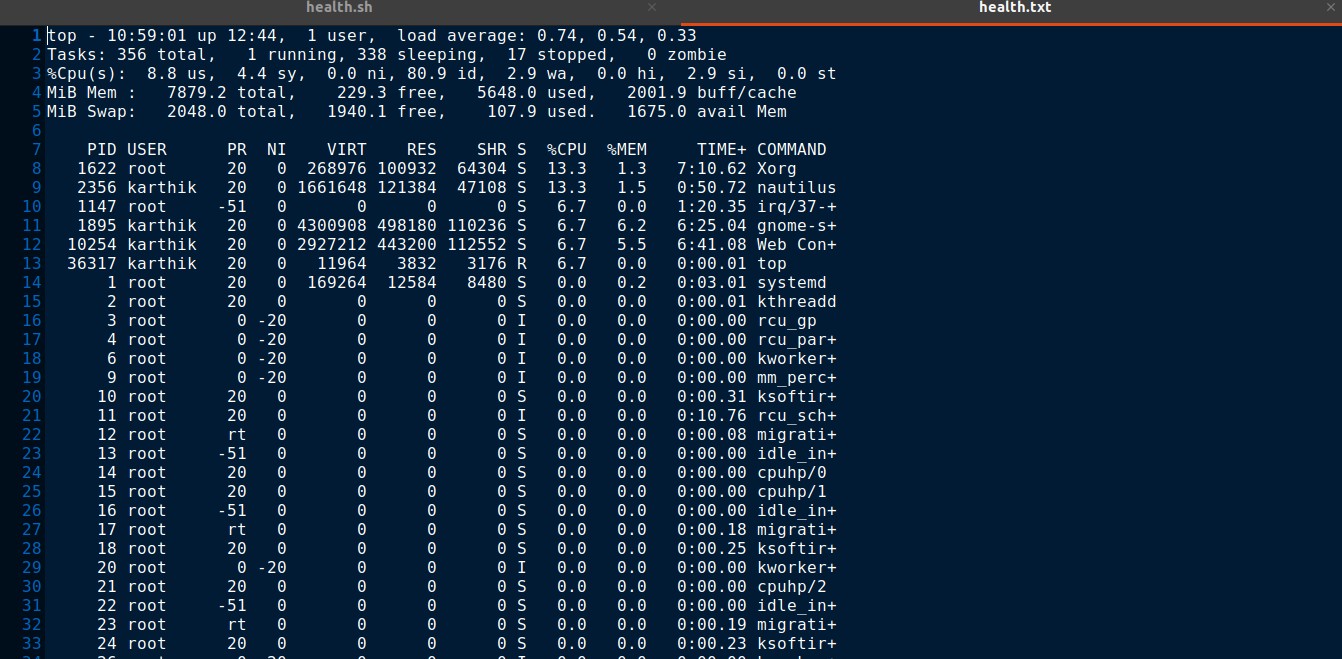
free---memory usage in the system

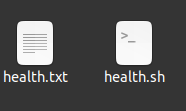
netstat---network info and socket connected info vmstat—virtual mem and cache info

Outputs:

Code:





files: health.sh and health.txt

In crontab



1. Script 2

Command:1

Rev command in Linux:

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

## rev -h ---Help

*rev -V ---Version Number*

**Output:**





## **Command:2**

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

## Output:

## 

## 

***Command:3***

***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’

**Output:**

## 

## 

## 

## 

## 

Additional Question:

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

## **Code for writer**

## #include <iostream>

## #include <sys/ipc.h>

## #include <sys/shm.h>

## #include <stdio.h>

## using namespace std;

## int main() {

## // ftok to generate unique key

## key\_t key = ftok("bhaveshfile",65);

## // shmget returns an identifier in shmid

## int shmid = shmget(key,1024,0666|IPC\_CREAT);

## // shmat to attach to shared memory

## char \*str = (char\*) shmat(shmid,(void\*)0,0);

## printf("Write Data : ");

## gets(str);

## printf("Data written in memory: %s\n",str);

## //detach from shared memory shmdt(str);

## return 0; }

*Code for reader:*

#include <iostream>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <stdio.h>

using namespace std;

int main() {

// ftok to generate unique key

key\_t key = ftok("bhaveshfile",65);

// shmget returns an identifier in shmid

int shmid = shmget(key,1024,0666|IPC\_CREAT);

// shmat to attach to shared memory

char \*str = (char\*) shmat(shmid,(void\*)0,0);

printf("Data read from memory: %s\n",str);

//detach from shared memory shmdt(str);

// destroy the shared memory shmctl(shmid,IPC\_RMID,NULL);

return 0;

}

Output:







