**Linux**

**Programming**

Sharon Deborah L

17mis1170

**Practice Lab Session**

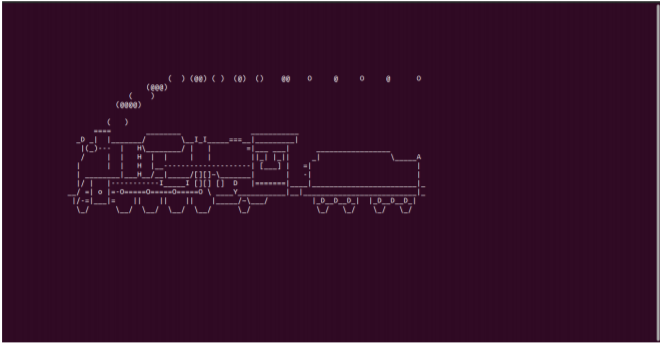
Also check out my GitHub here:

1. Trying out the command :

apt-get install sl

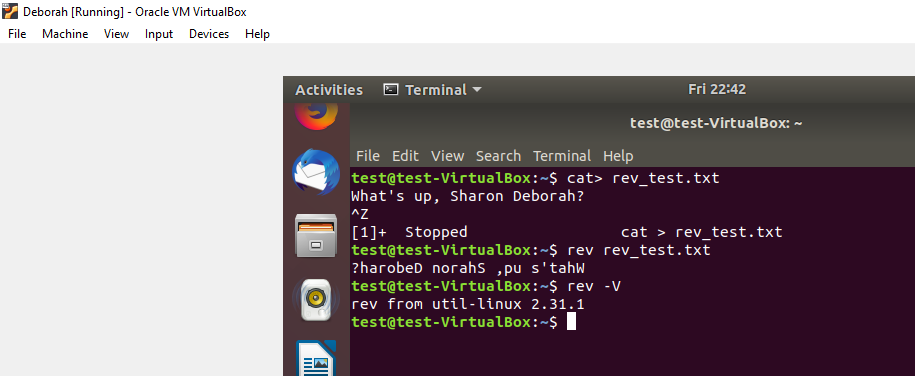
sl

Output:



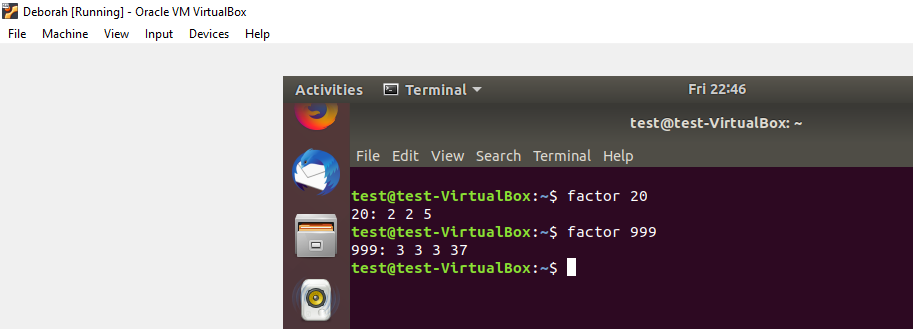
1. Rev command in Linux

Output:



1. The factor command in Linux is used to print the prime factors.

Output :

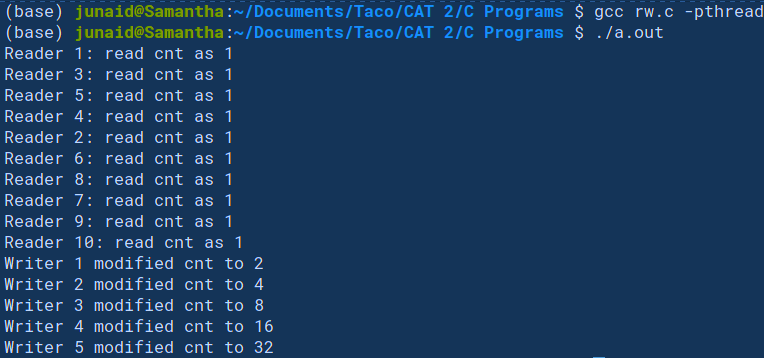


1. Solving the reader writer problem using semaphores

CODE :

|  |  |
| --- | --- |
|  | #include<pthread.h> |
|  | #include <semaphore.h> |
|  | #include <stdio.h> |
|  |  |
|  | sem\_t wrt; |
|  | pthread\_mutex\_t mutex; |
|  | int cnt = 1; |
|  | int numreader = 0; |
|  |  |
|  | void \*writer(void \*wno) |
|  | { |
|  | sem\_wait(&wrt); |
|  | cnt = cnt\*2; |
|  | printf("Writer %d modified cnt to %d\n",(\*((int \*)wno)),cnt); |
|  | sem\_post(&wrt); |
|  |  |
|  | } |
|  | void \*reader(void \*rno) |
|  | { |
|  | // Reader acquire the lock before modifying numreader |
|  | pthread\_mutex\_lock(&mutex); |
|  | numreader++; |
|  | if(numreader == 1) { |
|  | sem\_wait(&wrt); // If this id the first reader then it will block the writer |
|  | } |
|  | pthread\_mutex\_unlock(&mutex); |
|  | // Reading Section |
|  | printf("Reader %d: read cnt as %d\n",\*((int \*)rno),cnt); |
|  |  |
|  | // Reader acquire the lock before modifying numreader |
|  | pthread\_mutex\_lock(&mutex); |
|  | numreader--; |
|  | if(numreader == 0) { |
|  | sem\_post(&wrt); // If this is the last reader, it will wake up the writer. |
|  | } |
|  | pthread\_mutex\_unlock(&mutex); |
|  | } |
|  |  |
|  | int main() |
|  | { |
|  |  |
|  | pthread\_t read[10],write[5]; |
|  | pthread\_mutex\_init(&mutex, NULL); |
|  | sem\_init(&wrt,0,1); |
|  |  |
|  | int a[10] = {1,2,3,4,5,6,7,8,9,10}; //Just used for numbering the producer and consumer |
|  |  |
|  | for(int i = 0; i < 10; i++) { |
|  | pthread\_create(&read[i], NULL, (void \*)reader, (void \*)&a[i]); |
|  | } |
|  | for(int i = 0; i < 5; i++) { |
|  | pthread\_create(&write[i], NULL, (void \*)writer, (void \*)&a[i]); |
|  | } |
|  |  |
|  | for(int i = 0; i < 10; i++) { |
|  | pthread\_join(read[i], NULL); |
|  | } |
|  | for(int i = 0; i < 5; i++) { |
|  | pthread\_join(write[i], NULL); |
|  | } |
|  |  |
|  | pthread\_mutex\_destroy(&mutex); |
|  | sem\_destroy(&wrt); |
|  |  |
|  | return 0; |
|  |  |
|  | } |

Output:



1. Now the HOT one

Coding part:

vmstat 1200 > vmstat1.data

filename= "/home/Documents/vmstat1.data"

tail -f $filename |

while read $line do

if [ (cat vmstat1.data | grep "swap")>0 ]

then

echo "Consumed a lot of memory"> swap.txt

fi

if [ (cat vmstat1.data | grep "r")>1 ]

then

echo "waiting to execute"> runqueue.txt

fi

if [ (cat vmstat1.data | grep "cpu")>1000 ]

then

echo "cpu usage is more"> cpu.txt

fi

End