

## LINUX PROGRAMMING

NAME: SAMYUGDHA.S

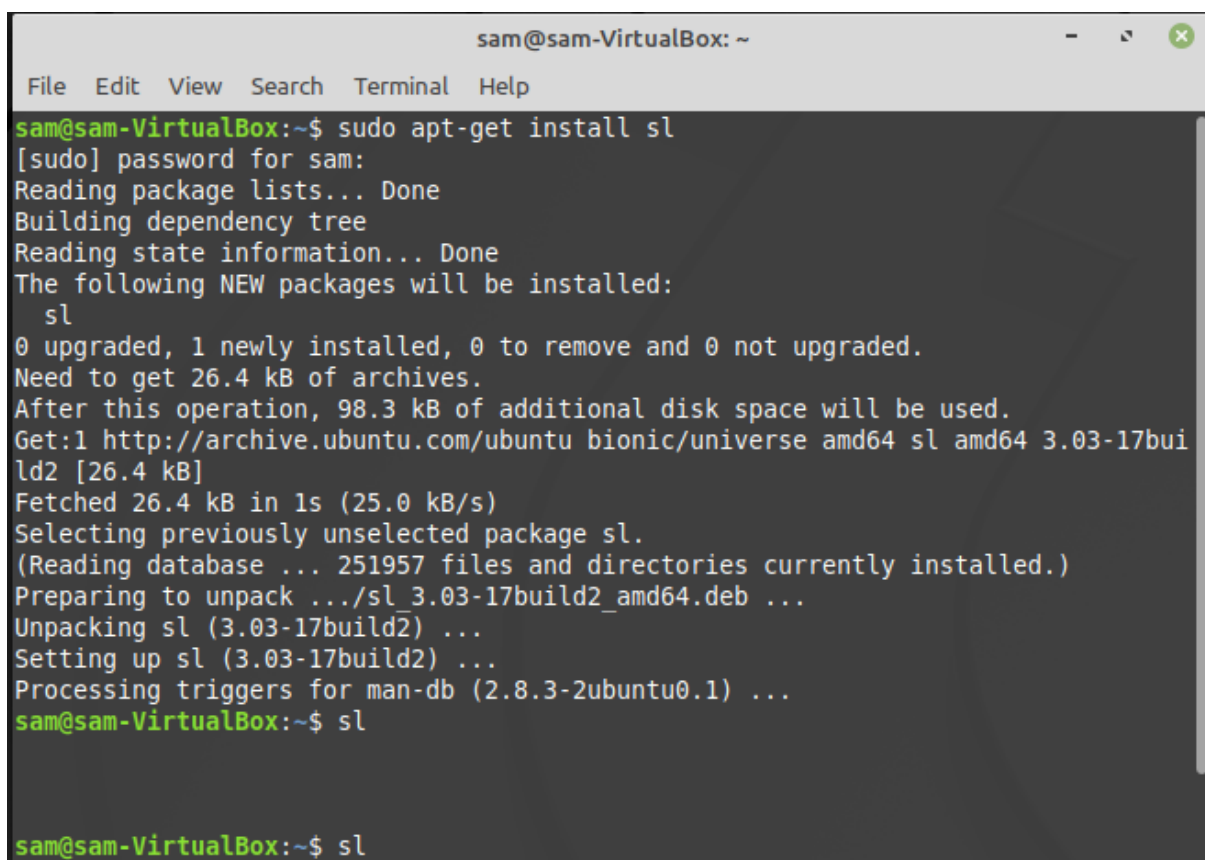
REGNO: 17MIS1118

### 1. sl

sudo apt-get install sl

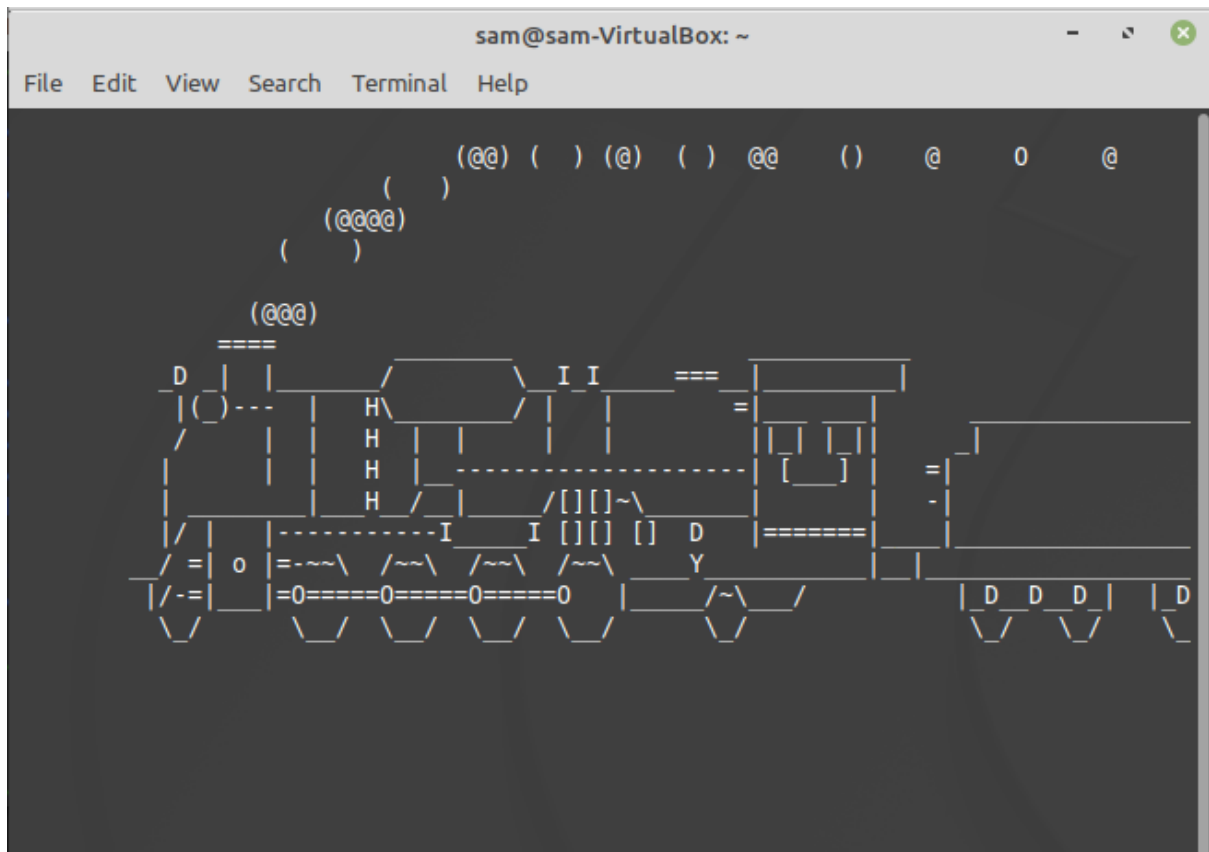
sl.sh

sl

A screenshot of a terminal window titled 'sam@sam-VirtualBox: ~'. The terminal shows the command 'sudo apt-get install sl' being executed. The output includes the password prompt, package list reading, dependency tree building, and state information reading. It then lists the new packages to be installed: 'sl'. The terminal shows the disk space requirements and the download of the package from the Ubuntu archive. Finally, it shows the package being unpacked and set up. The prompt returns to 'sam@sam-VirtualBox:~\$' and the user enters 'sl' again.

```
sam@sam-VirtualBox:~$ sudo apt-get install sl
[sudo] password for sam:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 26.4 kB of archives.
After this operation, 98.3 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 sl amd64 3.03-17bui
ld2 [26.4 kB]
Fetched 26.4 kB in 1s (25.0 kB/s)
Selecting previously unselected package sl.
(Reading database ... 251957 files and directories currently installed.)
Preparing to unpack .../sl 3.03-17build2_amd64.deb ...
Unpacking sl (3.03-17build2) ...
Setting up sl (3.03-17build2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
sam@sam-VirtualBox:~$ sl

sam@sam-VirtualBox:~$ sl
```



2.

Commands:

**rev**

**Factor**

**yes**

**sa.sh**

rev

factor

yes

```
sam@sam-VirtualBox:~/Desktop$ chmod +x sa.sh
sam@sam-VirtualBox:~/Desktop$ ./sa.sh
sam
mas
river
revir
linux
xunil

```

```
sam@sam-VirtualBox: ~  
File Edit View Search Terminal Help  
sam@sam-VirtualBox:~$ rev  
sam  
mas  
linux  
xunil  
^Csam@sam-VirtualBox:~$ factor  
78  
78: 2 3 13  
56  
56: 2 2 2 7  
25  
25: 5 5  
^C  
sam@sam-VirtualBox:~$ yes  
y  
y  
y  
y  
y  
y  
y  
y  
y  
y  
y  
y
```

3.

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.

```
vmstat 1200 > ss.data
```

```
filename= "/home/sam/Desktop/ss.data"
```

```
tail -f $filename |
```

```
while read $line do
```

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

```
then
```

```
    echo "some rogue process has consumed massive amounts of memory"> swap.txt
```

```
fi
```

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

```
then
```

```
    echo "some process are waiting to execute"> runqueue.txt
```

```
fi
```

```

if [ (cat ss.data | grep "cpu")>1000 ]
then
echo "cpu usage is more"> cpu.txt
fi
End

```

```

sam@sam-VirtualBox:~/Desktop$ chmod +x sys_health.sh
sam@sam-VirtualBox:~/Desktop$ ./sys_health.sh
^Csam@sam-VirtualBox:~/Desktop$

```

sa.sh x sys\_health.sh x ss.data x

procs		-----memory-----				---swap---		-----io----		-system--		-----cpu-----				
r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st
4	0	0	2487972	54080	827256	0	0	208	55	164	634	12	1	84	2	0

## GITHUB LINK:

[https://github.com/samyugdhas/sl\\_system\\_health\\_monitor](https://github.com/samyugdhas/sl_system_health_monitor)