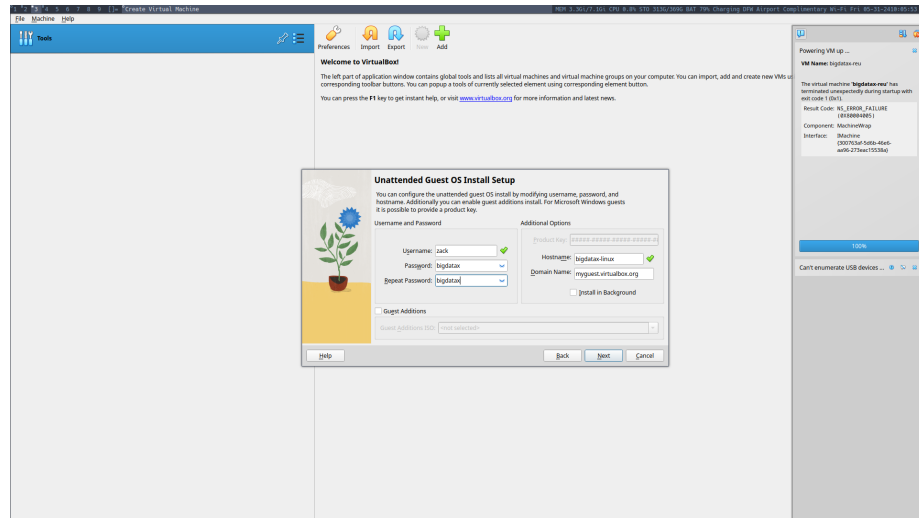


Linux Assignment

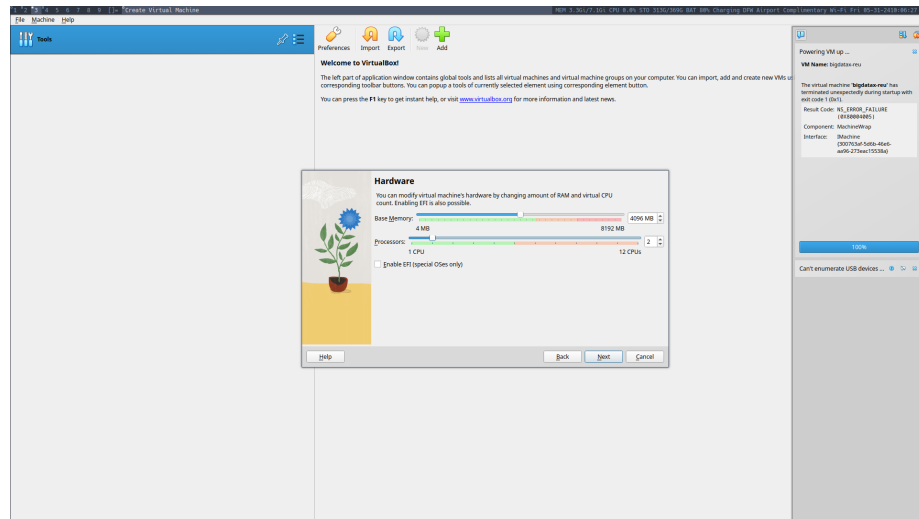
Zack Murry

I. Virtual Machine Setup

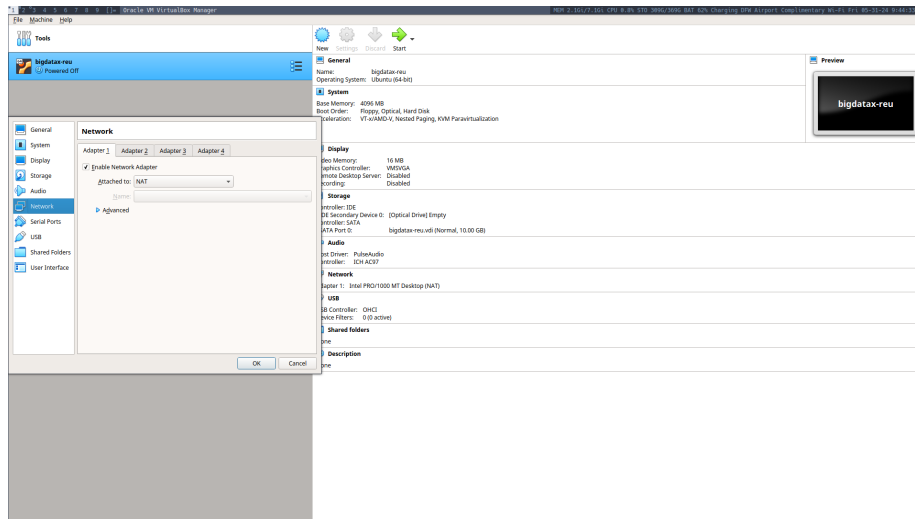
Create an Ubuntu 20.04 server virtual machine in Virtualbox. Configure its username and password.



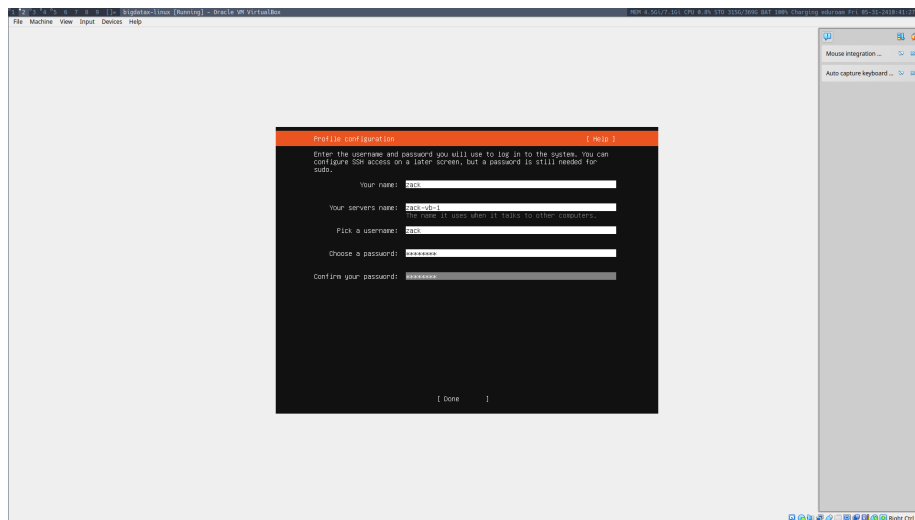
Allocate 4GB RAM and 2 CPU cores to the VM.



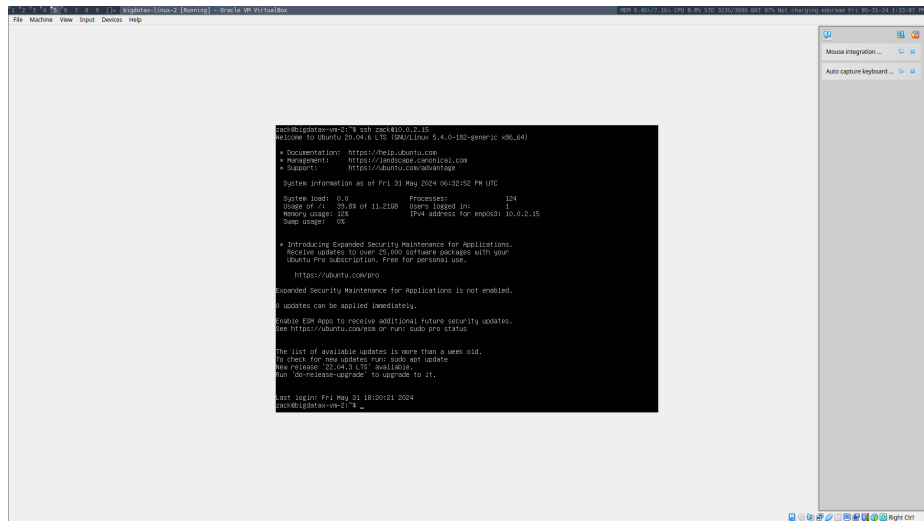
Allow network connections on the virtual machine.



Configure the username and password inside the Linux host.



Configure the firewall to deny all incoming connections besides SSH (port 22) using UFW.



```
zack@lispstar-vb-11:~$ ssh zack@10.0.2.15
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-102-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Fri 31 May 2024 06:32:52 PM UTC

System load: 0.0          Processes: 124
Usage of /: 39.8% of 11.2TB  Users logged in: 1
Memory usage: 11%        IP address for enp0s3: 10.0.2.15
Swap usage: 0%

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.
   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

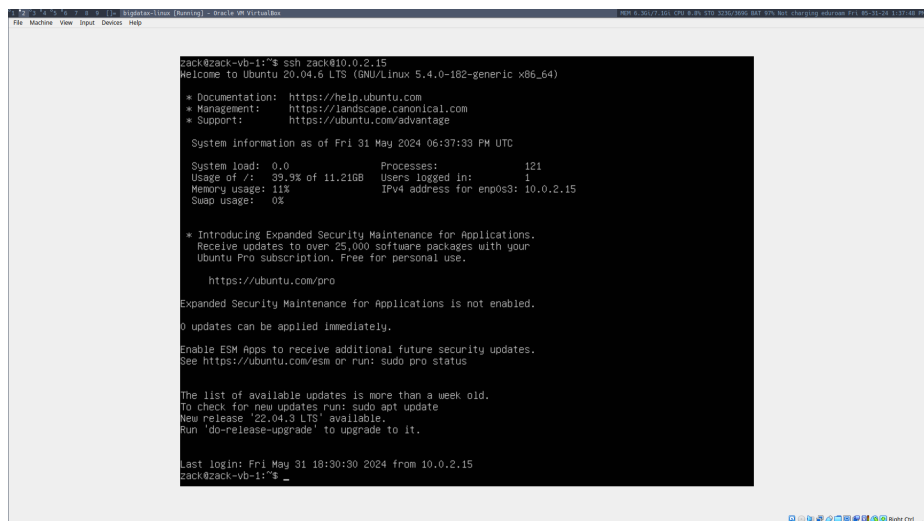
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri May 31 18:20:21 2024
zack@lispstar-vb-11:~$
```

II. Common Linux Commands

ssh

SSH enables secure shell-based connections between devices.



```
zack@zack-vb-11:~$ ssh zack@10.0.2.15
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-102-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Fri 31 May 2024 06:37:33 PM UTC

System load: 0.0          Processes: 121
Usage of /: 39.9% of 11.2TB  Users logged in: 1
Memory usage: 11%        IP address for enp0s3: 10.0.2.15
Swap usage: 0%

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.
   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri May 31 18:30:30 2024 from 10.0.2.15
zack@zack-vb-11:~$
```

ssh-keygen

SSH keygen is a simple interface for creating SSH keys.

```
zack@zack-vb-1:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/zack/.ssh/id_rsa): /home/zack/.ssh/id_test
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/zack/.ssh/id_test
Your public key has been saved in /home/zack/.ssh/id_test.pub
The key fingerprint is:
SHA256:pbno9Ph0ndabvFR425Xwtf22/h/RhxGI6M8xSHkv+U zack@zack-vb-1
The key's randomart image is:
[RSA 3072]-----+
|
|   = o
|   o + = .
|   * o o
|   S . + . .
|   oo . . o
| . = . . + . . +
| .800+00.0 . . + E+
|==00++0. .0. . 0sB
+----[SHA256]-----+
zack@zack-vb-1:~$
```

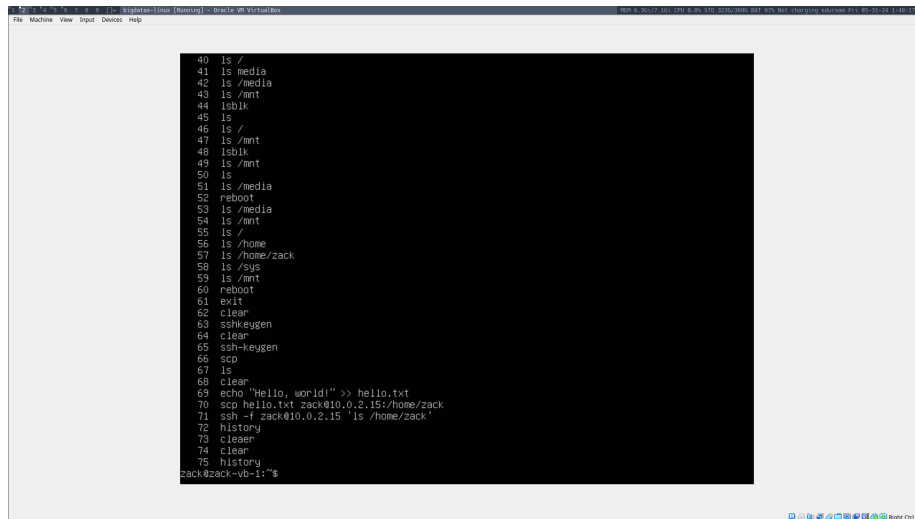
scp

scp allows the secure copying of files to a remote host.

```
zack@zack-vb-1:~$ echo "hello, world!" >> hello.txt
zack@zack-vb-1:~$ scp hello.txt zack@10.0.2.15:/home/zack
hello.txt                                100% 14   63.5KB/s   00:00
zack@zack-vb-1:~$ ssh -f zack@10.0.2.15 'ls /home/zack'
zack@zack-vb-1:~$ hello.txt
zack@zack-vb-1:~$
```

history

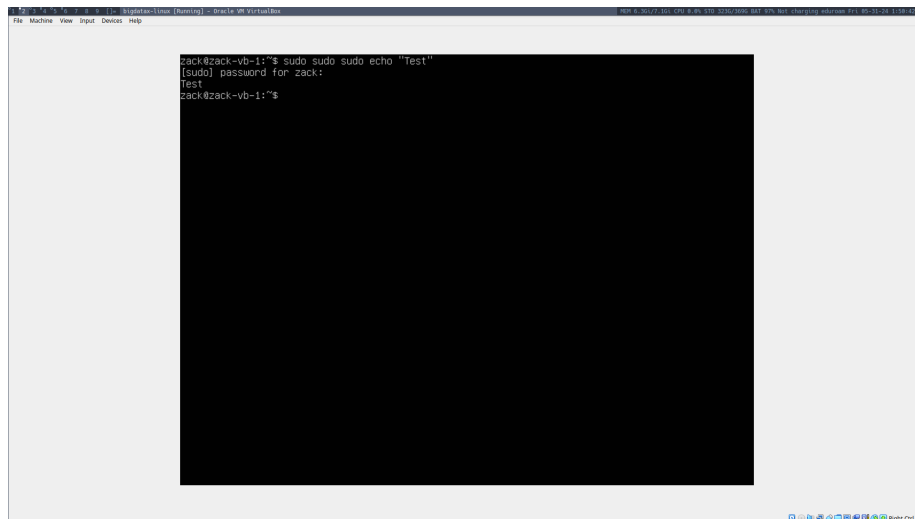
The **history** command lists all other commands that have been run in the current terminal session.



```
40 ls /
41 ls media
42 ls /media
43 ls /mnt
44 lsblk
45 ls
46 ls /
47 ls /mnt
48 lsblk
49 ls /mnt
50 ls
51 ls /media
52 reboot
53 ls /media
54 ls /mnt
55 ls /
56 ls /home
57 ls /home/zack
58 ls /sys
59 ls /mnt
60 reboot
61 exit
62 clear
63 ssh-keygen
64 clear
65 ssh-keygen
66 scp
67 ls
68 clear
69 echo "hello, world!" >> hello.txt
70 scp hello.txt zack@10.0.2.15:/home/zack
71 ssh -f zack@10.0.2.15 'ls /home/zack'
72 history
73 clear
74 clear
75 history
zack@zack-vb-l:~$
```

sudo

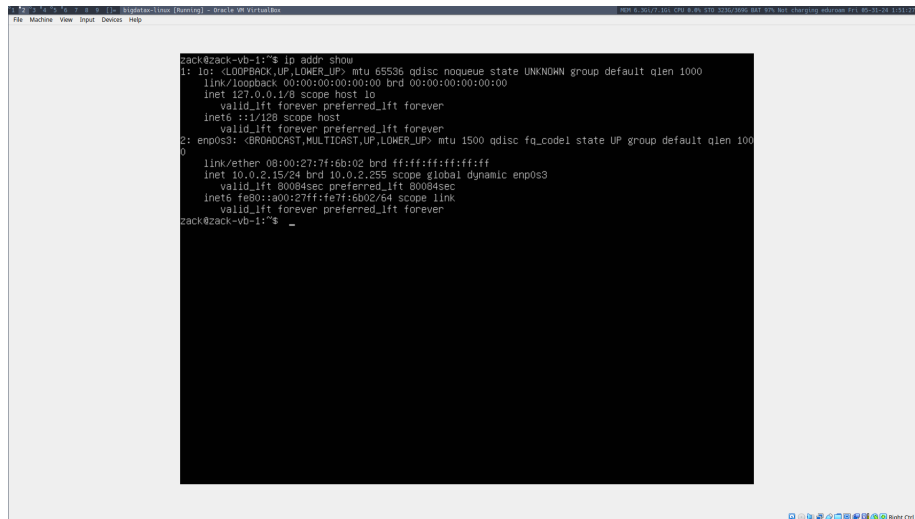
The **sudo** command allows users to run commands with elevated privileges.



```
zack@zack-vb-l:~$ sudo echo "Test"
[sudo] password for zack:
Test
zack@zack-vb-l:~$
```

ip

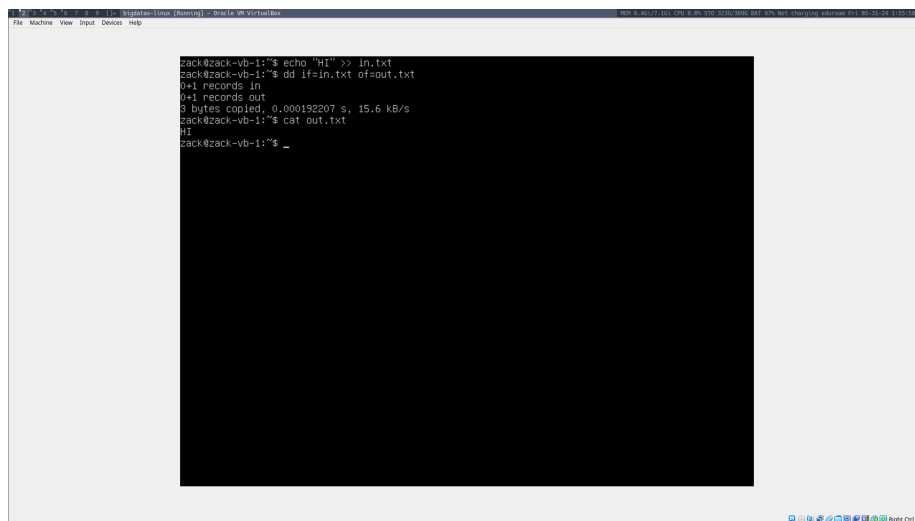
The **ip** command allows interfacing with the system's network devices.



```
zack@zack-vb-1:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BRIDGE,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ff:6b:02 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 80004sec preferred_lft 80004sec
    inet6 fe80::a00:27ff:fe7f:6b02/64 scope link
        valid_lft forever preferred_lft forever
zack@zack-vb-1:~$ _
```

dd

The **dd** command provides an interface for copying and deleting files and from disk devices.



```
zack@zack-vb-1:~$ echo "HI" >> in.txt
zack@zack-vb-1:~$ dd if=in.txt of=out.txt
0+1 records in
0+1 records out
3 bytes copied, 0.000192207 s, 15.6 kB/s
zack@zack-vb-1:~$ cat out.txt
HI
zack@zack-vb-1:~$ _
```

fdisk

The **fdisk** command allows the manipulation of disk partitions.

```
zack@zack-vb-1:~$ sudo fdisk -l | head -30
Disk /dev/loop0: 63.29 MiB, 66353296 bytes, 129608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop1: 91.05 MiB, 96320864 bytes, 188072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 49.86 MiB, 52260864 bytes, 102072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E695FE60-44E8-40F7-9815-4901508F2A32

Device      Start      End  Sectors  Size Type
/dev/sda1    2048      4095     2048    1M BIOS boot
zack@zack-vb-1:~$
```

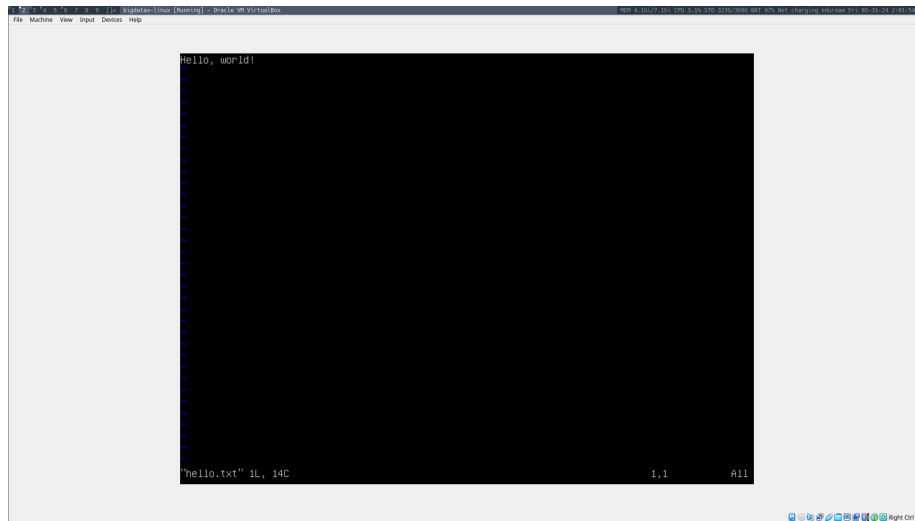
apt

The **apt** command provides a uniform interface for managing packages.

```
zack@zack-vb-1:~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 256 kB in 1s (329 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
57 packages can be upgraded. Run 'apt list --upgradable' to see them.
zack@zack-vb-1:~$
```

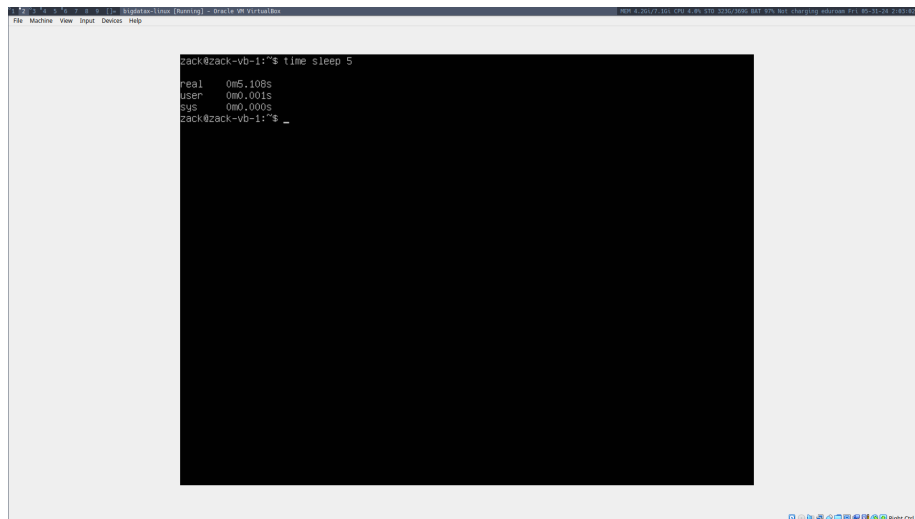
vi

The **vi** command opens an interactive text editor.



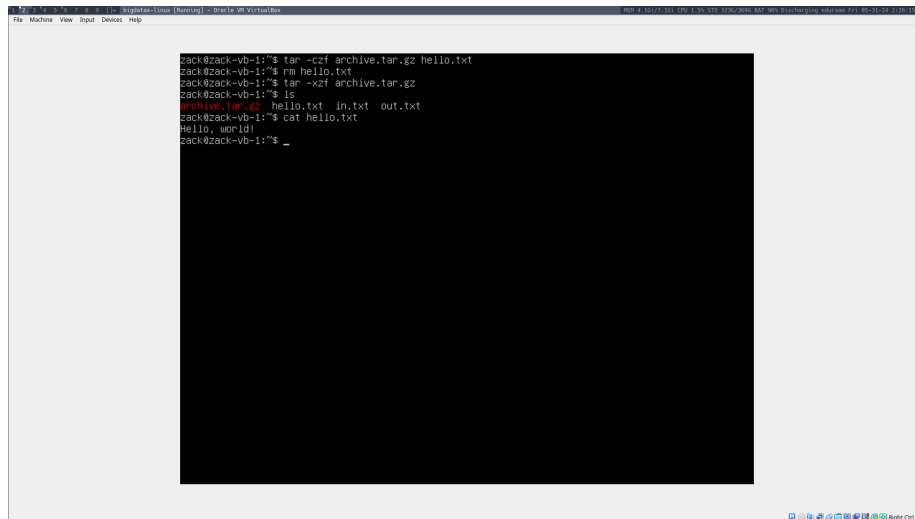
time

The **time** command measures the execution time of a given command.



tar

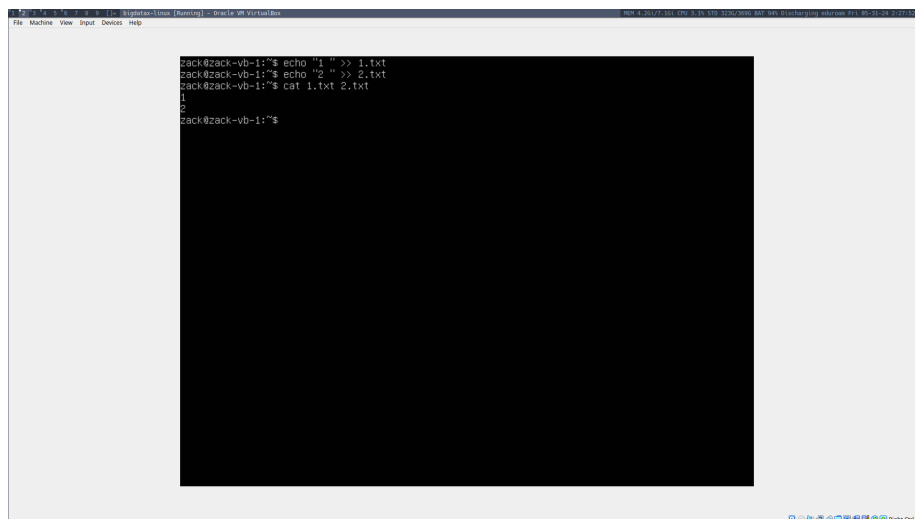
The **tar** command allows for compressing and decompressing files and directories.



```
zack@zack-vb-1:~$ tar -czf archive.tar.gz hello.txt
zack@zack-vb-1:~$ rm hello.txt
zack@zack-vb-1:~$ tar -xzf archive.tar.gz
zack@zack-vb-1:~$ ls
archive.tar.gz  hello.txt  in.txt  out.txt
zack@zack-vb-1:~$ cat hello.txt
Hello, world!
zack@zack-vb-1:~$ _
```

cat

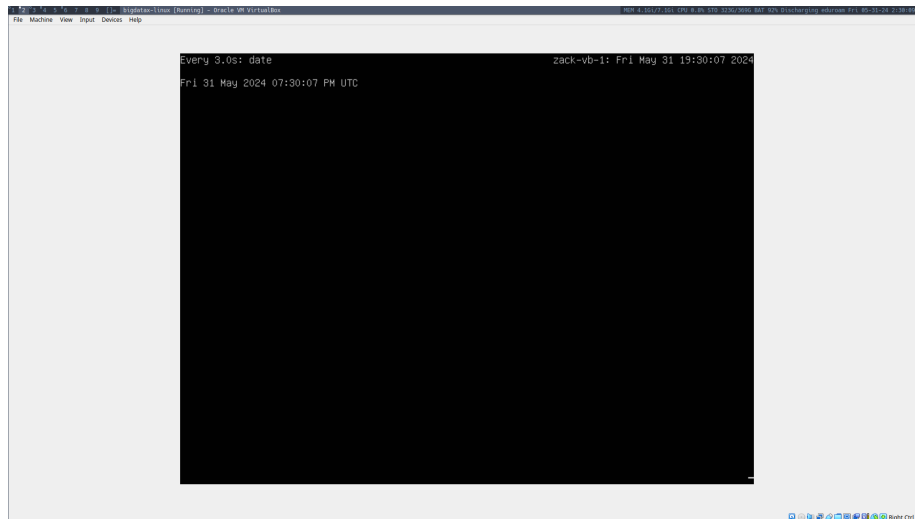
The **cat** command conCATenates files to the standard output.



```
zack@zack-vb-1:~$ echo "1" >> 1.txt
zack@zack-vb-1:~$ echo "2" >> 2.txt
zack@zack-vb-1:~$ cat 1.txt 2.txt
1
2
zack@zack-vb-1:~$
```

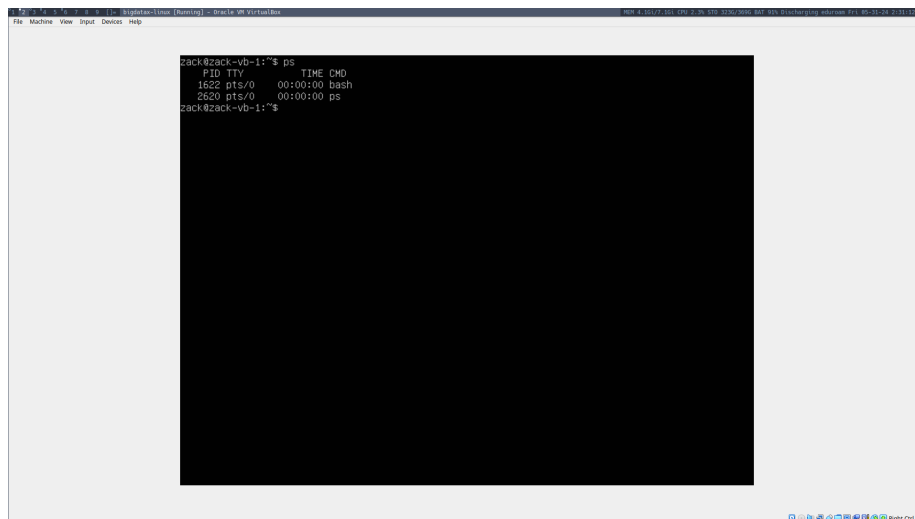
watch

The **watch** command executes a program at a given interval.



ps

The **ps** command lists the active processes.



top

The **top** command displays the resources used by every active process.

```
Tasks: 107 total, 1 running, 106 sleeping, 0 stopped, 0 zombie
Rtzu(s): 0.0 us, 0.2 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1971.4 total, 1126.7 free, 165.5 used, 679.2 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 1655.2 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        20   0 102604 11600 8532 S   0.3   0.6   0:01.62 systemd
 1621 zack        20   0 13940 5776 4312 S   0.3   0.3   0:00.32 sshd
    2 root        20   0 0 0 0 S   0.0   0.0   0:00.00 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_pap_gp
    6 root        0 -20 0 0 0 I   0.0   0.0   0:00.00 kworker/0:0H
    8 root        0 -20 0 0 0 I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root        20   0 0 0 0 S   0.0   0.0   0:00.06 ksoftirqd/0
   10 root        20   0 0 0 0 I   0.0   0.0   0:02.07 rcu_sched
   11 root        rt   0 0 0 0 S   0.0   0.0   0:00.02 migration/0
   12 root       -51   0 0 0 0 S   0.0   0.0   0:00.00 idle_inject/0
   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_inject/1
   17 root        rt   0 0 0 0 S   0.0   0.0   0:00.27 migration/1
   18 root        20   0 0 0 0 S   0.0   0.0   0:00.06 ksoftirqd/1
   20 root        0 -20 0 0 0 I   0.0   0.0   0:00.00 kworker/1:0H-kblockd
   21 root        20   0 0 0 0 S   0.0   0.0   0:00.00 kdevtmpfs
   22 root        0 -20 0 0 0 I   0.0   0.0   0:00.00 netns
   23 root        20   0 0 0 0 S   0.0   0.0   0:00.00 rcu_tasks_kthre
   24 root        20   0 0 0 0 S   0.0   0.0   0:00.00 kauditd
   25 root        20   0 0 0 0 S   0.0   0.0   0:00.00 khungtaskd
   26 root        20   0 0 0 0 S   0.0   0.0   0:00.00 oom_reaper
   27 root        0 -20 0 0 0 I   0.0   0.0   0:00.00 writelock
   28 root        20   0 0 0 0 S   0.0   0.0   0:00.00 kcompactd0
   29 root       25   5 0 0 0 S   0.0   0.0   0:00.00 ksmd
   30 root       39  19 0 0 0 S   0.0   0.0   0:00.00 khugepaged
   77 root        0 -20 0 0 0 I   0.0   0.0   0:00.00 kintegrityd
   78 root        0 -20 0 0 0 I   0.0   0.0   0:00.00 kblockd
   79 root        0 -20 0 0 0 I   0.0   0.0   0:00.00 blkcg_punt_bio

zack@zack-v0-l1:~$
```

htop

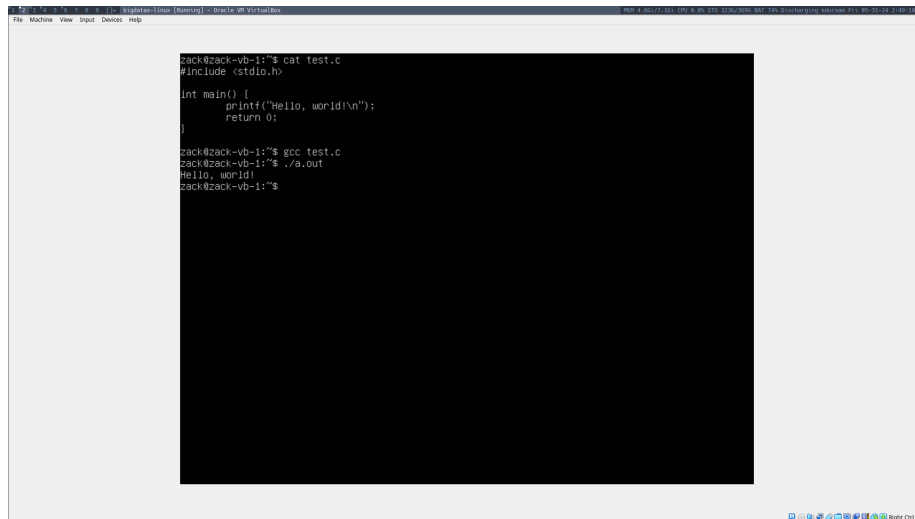
The `htop` command is an alternative to the `top` command with an improved user interface.

```
Tasks: 30, 32 thr; 1 running
Load average: 0.00 0.01 0.00
Uptime: 02:22:57

  PID USER      PRI  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  Command
 2535 zack        20   0 8024 3880 3148 R   0.0   0.2   0:00.53 htop
 669 root        20   0 854M 38096 19092 S   0.0   1.9   0:01.70 /usr/lib/snapd/snapd
 565 root        RT   0 273M 18000 8208 S   0.0   0.9   0:00.85 /sbin/multitpath -d -s
1553 zack        20   0 12016 6388 5688 S   0.0   0.3   0:00.58 ssh zack@10.0.2.15
 401 root        20   0 23160 6672 4076 S   0.0   0.3   0:00.98 /lib/systemd/systemd-udevd
 642 systemd-r-  20   0 24680 12276 8208 S   0.0   0.6   0:00.11 /lib/systemd/systemd-resolved
1621 zack        20   0 13940 5776 4312 S   0.0   0.3   0:00.33 sshd: zack@pts/0
 663 root        20   0 81828 3820 3524 S   0.0   0.2   0:00.21 /usr/sbin/irqbalance --foreground
 654 root        20   0 230M 7364 6536 S   0.0   0.4   0:00.16 /usr/lib/accounts-service/accounts-d
 803 root        20   0 854M 38096 19092 S   0.0   1.9   0:00.09 /usr/lib/snapd/snapd
 569 root        RT   0 273M 18000 8208 S   0.0   0.9   0:00.40 /sbin/multitpath -d -s
 801 root        20   0 854M 38096 19092 S   0.0   1.9   0:00.10 /usr/lib/snapd/snapd
 700 root        20   0 230M 7364 6536 S   0.0   0.4   0:00.14 /usr/lib/accounts-service/accounts-d
    1 root        20   0 100M 11600 8532 S   0.0   0.6   0:01.62 /sbin/init maybe-ubiquiti
 365 root       19 -1 52152 17766 16716 S   0.0   0.9   0:00.29 /lib/systemd/systemd-journald
 567 root        RT   0 273M 18000 8208 S   0.0   0.9   0:00.06 /sbin/multitpath -d -s
 568 root        RT   0 273M 18000 8208 S   0.0   0.9   0:00.01 /sbin/multitpath -d -s
 570 root        RT   0 273M 18000 8208 S   0.0   0.9   0:00.00 /sbin/multitpath -d -s
 571 root        RT   0 273M 18000 8208 S   0.0   0.9   0:00.00 /sbin/multitpath -d -s
 608 systemd-t-  20   0 90880 6108 5328 S   0.0   0.3   0:00.00 /lib/systemd/systemd-timesyncd
 603 systemd-t-  20   0 90880 6108 5328 S   0.0   0.3   0:00.10 /lib/systemd/systemd-timesyncd
 640 systemd-r-  20   0 27264 7536 6664 S   0.0   0.4   0:00.08 /lib/systemd/systemd-networkd
 715 root        20   0 230M 7364 6536 S   0.0   0.4   0:00.00 /usr/lib/accounts-service/accounts-d
 657 root        20   0 6815 2768 2564 S   0.0   0.1   0:00.01 /usr/sbin/cron -f
 658 messagebu-  20   0 7572 4844 4060 S   0.0   0.2   0:00.11 /usr/bin/dbus-daemon --system --add
 678 root        20   0 81828 3820 3524 S   0.0   0.2   0:00.00 /usr/sbin/irqbalance --foreground
 665 root        20   0 23648 18284 10272 S   0.0   0.9   0:00.09 /usr/bin/python3 /usr/bin/networkd-
 679 root        20   0 7276 6812 4128 S   0.0   0.3   0:00.00 /usr/lib/python3.11/_localeid --new
```

gcc

The GNU C Compiler (`gcc`) is used to compile C programs into executable programs.



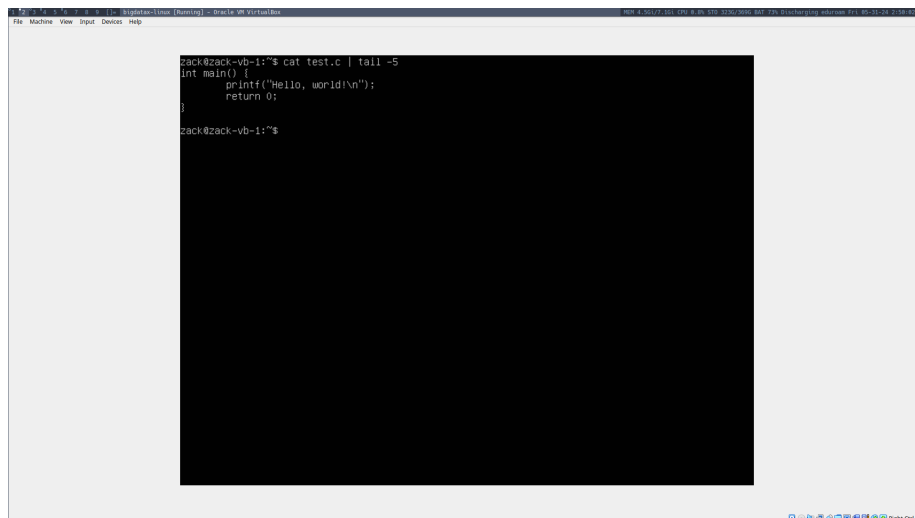
```
zack@zack-vb-1:~$ cat test.c
#include <stdio.h>

int main() {
    printf("Hello, world!\n");
    return 0;
}

zack@zack-vb-1:~$ gcc test.c
zack@zack-vb-1:~$ ./a.out
Hello, world!
zack@zack-vb-1:~$
```

tail

The **tail** command is used to retrieve the end of a file or stream.

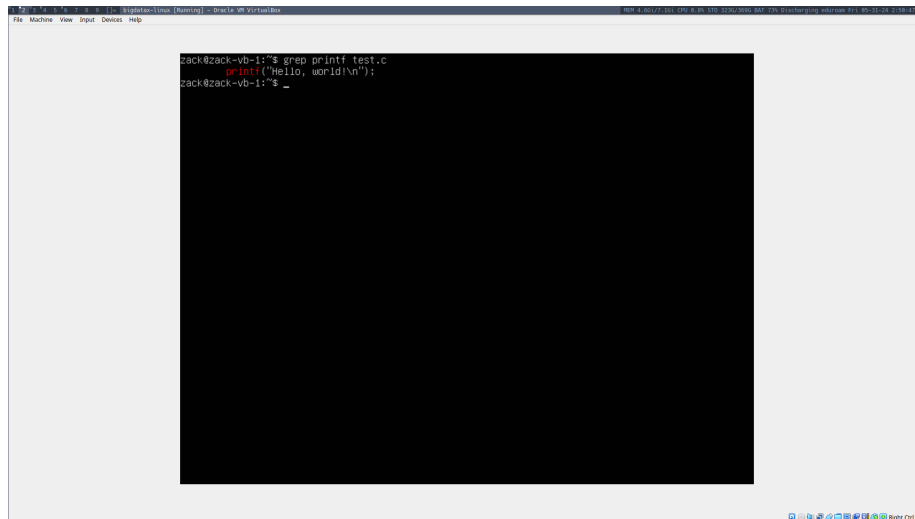


```
zack@zack-vb-1:~$ cat test.c | tail -5
int main() {
    printf("Hello, world!\n");
    return 0;
}

zack@zack-vb-1:~$
```

grep

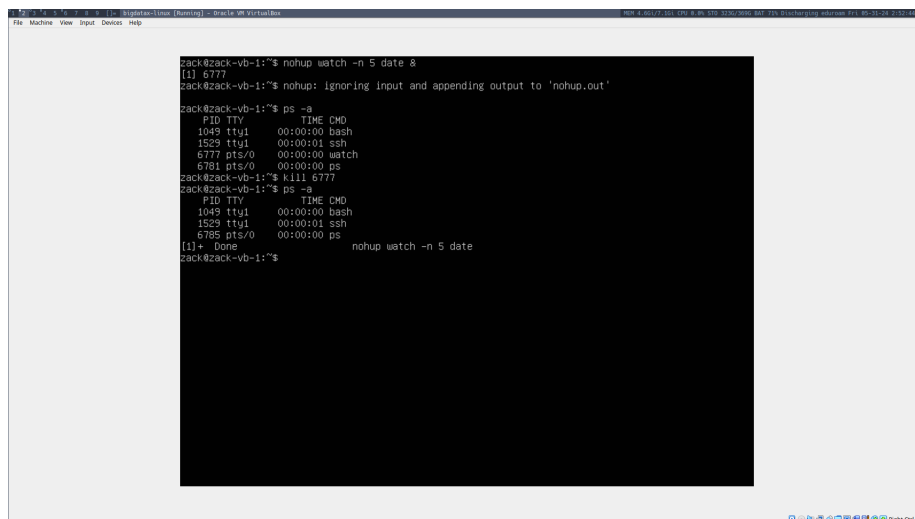
The **grep** (global regular expression print) command is used to search files via regular expressions.



```
zack@zack-vb-1:~$ gcc printf test.c
printf: ("Hello, world!\n");
zack@zack-vb-1:~$ _
```

kill

The `kill` command is used to end a process based on its ID.



```
zack@zack-vb-1:~$ nohup watch -n 5 date &
[1] 6777
zack@zack-vb-1:~$ nohup: Ignoring input and appending output to 'nohup.out'

zack@zack-vb-1:~$ ps -a
PID TTY          TIME CMD
1049 ttty1        00:00:00 bash
1529 ttty1        00:00:01 ssh
6777 pts/0      00:00:00 watch
6781 pts/0      00:00:00 ps
zack@zack-vb-1:~$ kill 6777
zack@zack-vb-1:~$ ps -a
PID TTY          TIME CMD
1049 ttty1        00:00:00 bash
1529 ttty1        00:00:01 ssh
6785 pts/0      00:00:00 ps
[1]+  Done                  nohup watch -n 5 date
zack@zack-vb-1:~$
```

killall

The `killall` command is used to end all processes with a given name.

```
zack@zack-vb-1:~$ nohup watch -n 5 date &
[1] 6793
zack@zack-vb-1:~$ nohup: ignoring input and appending output to 'nohup.out'

zack@zack-vb-1:~$ ps -a
PID TTY          TIME CMD
1049 ttty1        00:00:00 bash
1529 ttty1        00:00:01 ssh
6793 pts/0      00:00:00 watch
6797 pts/0      00:00:00 ps
zack@zack-vb-1:~$ killall watch
zack@zack-vb-1:~$ ps -a
PID TTY          TIME CMD
1049 ttty1        00:00:00 bash
1529 ttty1        00:00:01 ssh
6802 pts/0      00:00:00 ps
[1]+  Done                  nohup watch -n 5 date
zack@zack-vb-1:~$
```

du

The `du` command is used to view disk usage in a directory.

```
zack@zack-vb-1:~$ du
8      ./config/http
4      ./config/procps
16     ./config
28     ./ssh
4      ./cache
124    .
zack@zack-vb-1:~$
```

df

The `df` command is used to view total and available space on a filesystem.

```
zack@zack-vb-1:~$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
udev                  962572      0   962572    0% /dev
tmpfs                 201968    1096   200772    1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 11758760 4839056  6909336   41% /
tmpfs                 1009336      0   1009336    0% /dev/shm
tmpfs                  5120      0     5120    0% /run/lock
tmpfs                 1009336      0   1009336    0% /sys/fs/cgroup
/dev/loop2             51072    51072      0 100% /snap/snapd/18357
/dev/loop0             64896    64896      0 100% /snap/core20/1828
/dev/loop1             94080    94080      0 100% /snap/lxd/24061
/dev/sda2             1992552   111240   1780072    6% /boot
tmpfs                  201864      0   201864    0% /run/user/1000

zack@zack-vb-1:~$
```

screen

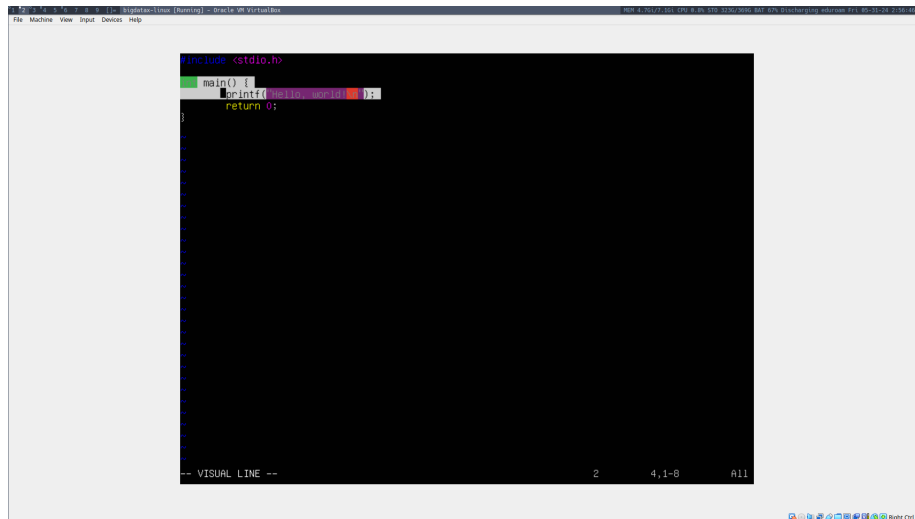
The **screen** command is used to concurrently run processes in the background, retrieve them, and kill them.

```
zack@zack-vb-1:~$ screen -ls
There is a screen on:
  6822.dts-0.zack-vb-1    (05/31/2024 07:55:35 PM)      (Detached)
1 Socket in /run/screen/S-zack.
zack@zack-vb-1:~$ screen -XS 6822 quit
zack@zack-vb-1:~$ screen -ls
No Sockets found in /run/screen/S-zack.

zack@zack-vb-1:~$
```

vim

The **vim** command is used to view and edit files interactively.

A screenshot of a code editor window titled "zack@zack-vb-1: ~ - Oracle VM VirtualBox". The editor shows a C program with the following code:

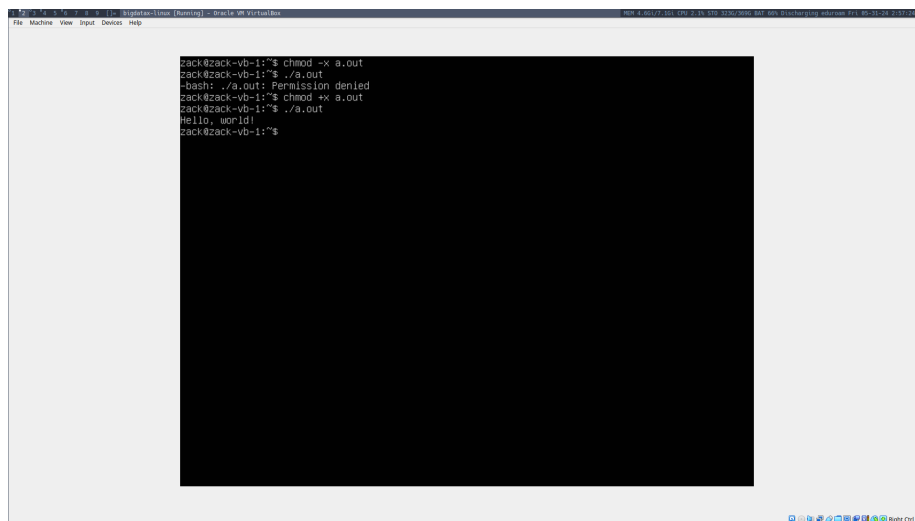
```
#include <stdio.h>

main() {
    printf("Hello, world!\n");
    return 0;
}
```

The code is syntax-highlighted. The editor has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The status bar at the bottom shows "2", "4,1-0", and "All".

chmod

The **chmod** command is used to change the access permissions for files or directories.

A screenshot of a terminal window titled "zack@zack-vb-1: ~ - Oracle VM VirtualBox". The terminal shows the following commands and output:

```
zack@zack-vb-1:~$ chmod -x a.out
zack@zack-vb-1:~$ ./a.out
-bash: ./a.out: Permission denied
zack@zack-vb-1:~$ chmod +x a.out
zack@zack-vb-1:~$ ./a.out
Hello, world!
zack@zack-vb-1:~$
```

The terminal has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The status bar at the bottom shows "2", "4,1-0", and "All".

chown

The **chown** command is used to change the owner of a file or directory.

```
zack@zack-vb-1:~$ sudo chown root a.out
zack@zack-vb-1:~$ ls -l
total 52
-rw-rw-r-- 1 zack zack   3 May 31 19:27 1.txt
-rw-rw-r-- 1 zack zack   3 May 31 19:27 2.txt
-rwxrwxr-x 1 root zack 16696 May 31 19:40 a.out
-rw-rw-r-- 1 zack zack  140 May 31 19:25 archive.tar.gz
-rw-rw-r-- 1 zack zack  14 May 31 18:42 hello.txt
-rw-rw-r-- 1 zack zack   3 May 31 18:55 in.txt
-rw-rw-r-- 1 zack zack 1163 May 31 19:53 nohup.out
-rw-rw-r-- 1 zack zack   3 May 31 18:55 out.txt
-rw-rw-r-- 1 zack zack   75 May 31 19:38 test.c
zack@zack-vb-1:~$
```

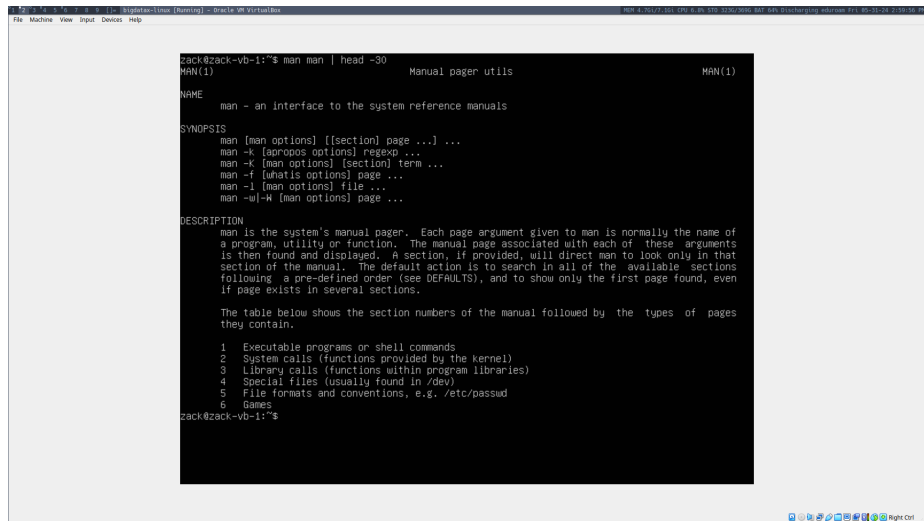
useradd

The `useradd` command is used to add a user account to a system.

```
zack@zack-vb-1:~$ sudo useradd new_user
zack@zack-vb-1:~$
```

man

The `man` command opens the manual of a given command.



```
zack@zack-vb-1:~$ man man | head -50
MAN(1)                                     Manual pager utils                                     MAN(1)

NAME
  man - an interface to the system reference manuals

SYNOPSIS
  man [man options] [[section] page ...] ...
  man -k [options options] regexp ...
  man -K [man options] [section] term ...
  man -f [whatis options] page ...
  man -l [man options] file ...
  man -w [-k [man options] page ...]

DESCRIPTION
  man is the system's manual pager.  Each page argument given to man is normally the name of
  a program, utility or function.  The manual page associated with each of these arguments
  is then found and displayed.  A section, if provided, will direct man to look only in that
  section of the manual.  The default action is to search in all of the available sections
  following a pre-defined order (see DEFAULTS), and to show only the first page found, even
  if page exists in several sections.

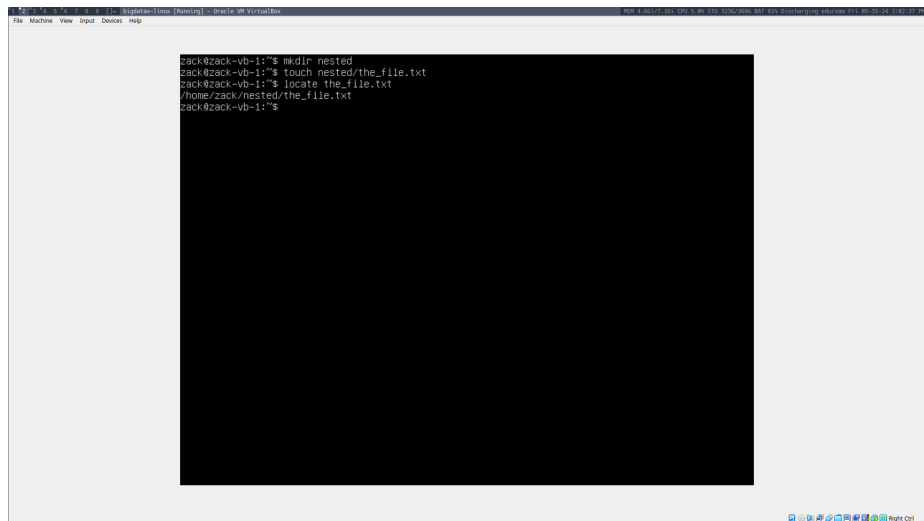
  The table below shows the section numbers of the manual followed by the types of pages
  they contain.

  1 Executable programs or shell commands
  2 System calls (functions provided by the kernel)
  3 Library calls (functions within program libraries)
  4 Special files (usually found in /dev)
  5 File formats and conventions, e.g. /etc/passwd
  6 Games

zack@zack-vb-1:~$
```

locate

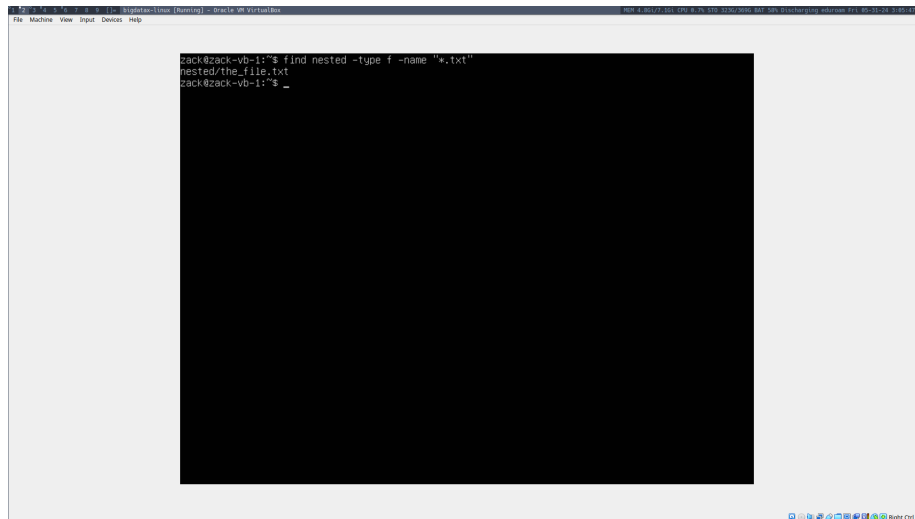
The **locate** command locates a file or directory in the filesystem by its name.



```
zack@zack-vb-1:~$ mkdir nested
zack@zack-vb-1:~$ touch nested/the_file.txt
zack@zack-vb-1:~$ locate the_file.txt
/home/zack/nested/the_file.txt
zack@zack-vb-1:~$
```

find

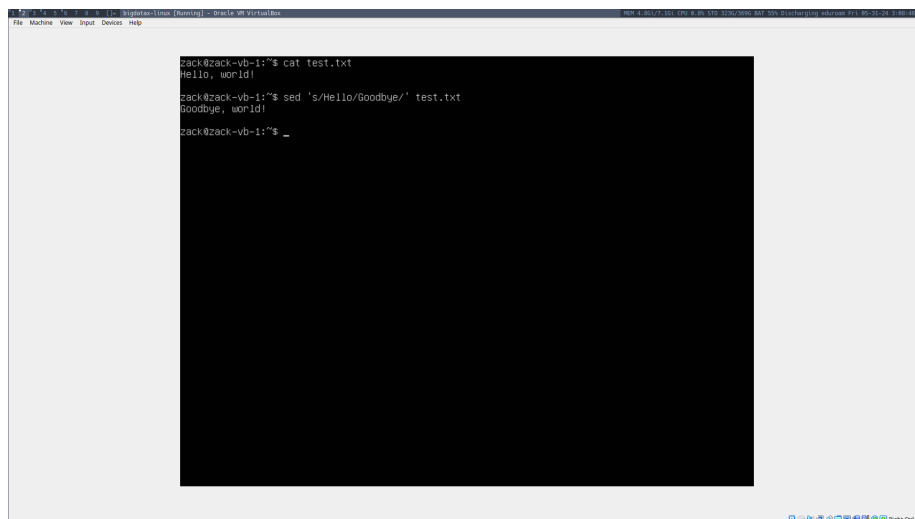
The **locate** command allows searching for files via their attributes, such as the name, size, or modification date of the files.



```
zack@zack-vb-1:~$ find nested -type f -name "*.txt"
nested/the_file.txt
zack@zack-vb-1:~$ _
```

sed

The **sed** command is used for substituting text in a file and outputting the new text to a stream.

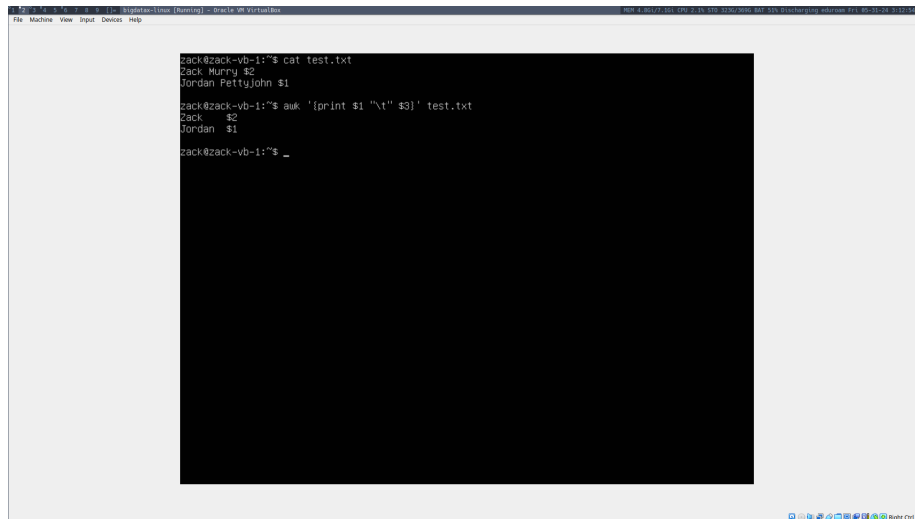


```
zack@zack-vb-1:~$ cat test.txt
Hello, world!

zack@zack-vb-1:~$ sed 's/Hello/Goodbye/' test.txt
Goodbye, world!
zack@zack-vb-1:~$ _
```

awk

The **awk** command is used to interface with a scripting language that is generally used for editing and filtering text.

A screenshot of a terminal window titled "Oracle VM VirtualBox". The terminal shows a shell script being executed. The script contains several lines of code, including a 'cat' command to display the contents of 'test.txt', an 'awk' command to process the file, and a 'print' statement. The output of the script is displayed in the terminal.

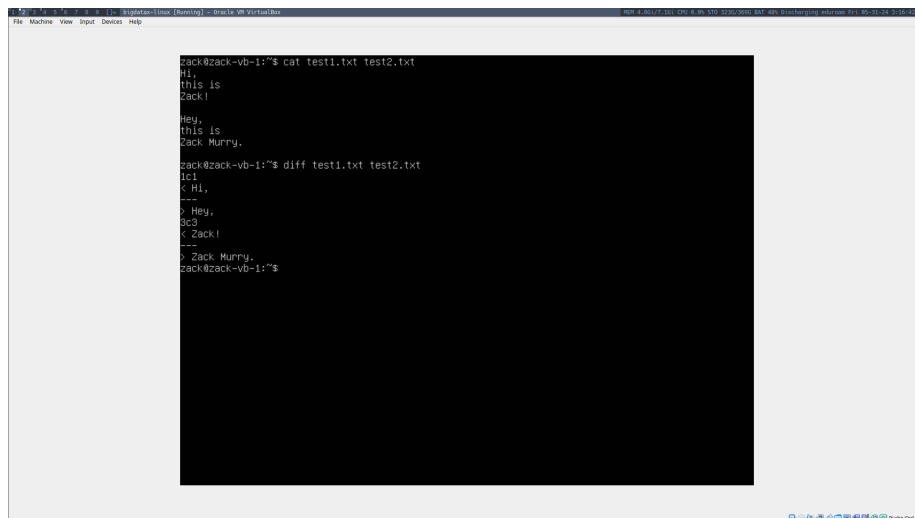
```
zack@zack-vb-1:~$ cat test.txt
Zack Munny $2
Jordan Pettyjohn $1

zack@zack-vb-1:~$ awk '{print $1 "\t" $3}' test.txt
Zack      $2
Jordan    $1

zack@zack-vb-1:~$ _
```

diff

The **diff** command is used to find the difference between files.

A screenshot of a terminal window titled "Oracle VM VirtualBox". The terminal shows the execution of the 'diff' command to compare two files, 'test1.txt' and 'test2.txt'. The output of the command is displayed in the terminal, showing the differences between the two files.

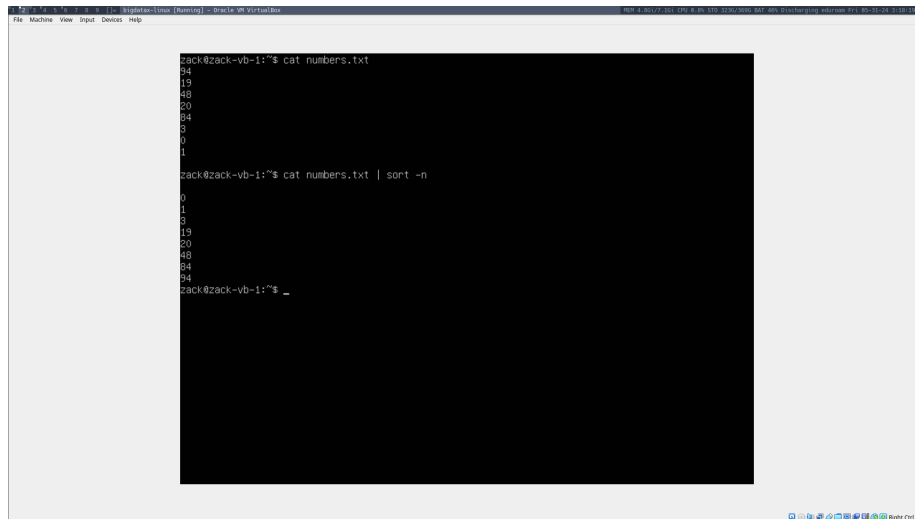
```
zack@zack-vb-1:~$ cat test1.txt test2.txt
Hi,
this is
Zack!

Hey,
this is
Zack Munny.

zack@zack-vb-1:~$ diff test1.txt test2.txt
1c1
< Hi,
---
> Hey,
2c2
< Zack!
---
> Zack Munny.
zack@zack-vb-1:~$
```

sort

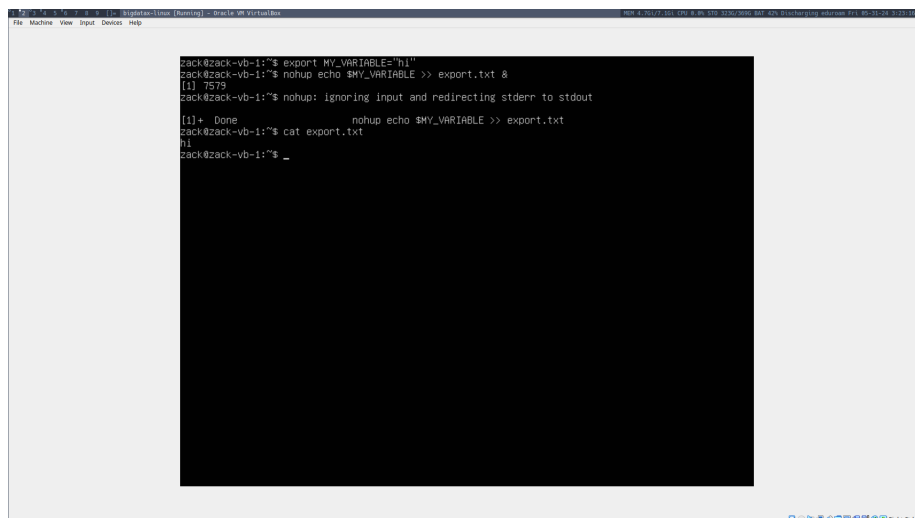
The **sort** command is used to sort data.



```
zack@zack-vb-1:~$ cat numbers.txt
94
19
48
20
94
9
0
1
zack@zack-vb-1:~$ cat numbers.txt | sort -n
0
1
9
19
20
48
94
94
zack@zack-vb-1:~$ _
```

export

The **export** command is used to pass environment variables, functions, and variables to child processes.



```
zack@zack-vb-1:~$ export MY_VARIABLE="hi"
zack@zack-vb-1:~$ nohup echo $MY_VARIABLE >> export.txt &
[1] 7579
zack@zack-vb-1:~$ nohup: ignoring input and redirecting stderr to stdout
[1]+  Done                  nohup echo $MY_VARIABLE >> export.txt
zack@zack-vb-1:~$ cat export.txt
hi
zack@zack-vb-1:~$ _
```

pwd

The **pwd** command is used to Print the Working Directory.


```
zack@zack-vb-1:~$ mount -l | head -30
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,nodev,noexec,relatime,size=962572k,nr_inodes=240643,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,nodev,noexec,relatime,gid=5,mode=600,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=201868k,mode=755)
/dev/mapper/ubuntu--vg-ubuntu--lv on / type ext4 (rw,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nodeslegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/bkio type cgroup (rw,nosuid,nodev,noexec,relatime,bkio)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=28,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=16183)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
tracets on /sys/kernel/tracing type tracets (rw,nosuid,nodev,noexec,relatime)
zack@zack-vb-1:~$
```

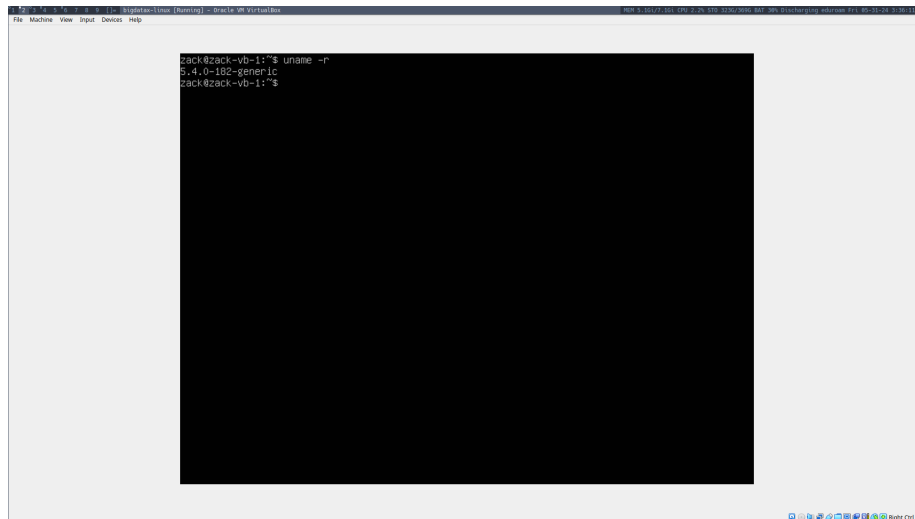
passwd

The `passwd` command is used to change a user's password.

```
zack@zack-vb-1:~$ passwd
Changing password for zack.
Current password:
New password:
Retype new password:
passwd: password updated successfully
zack@zack-vb-1:~$
```

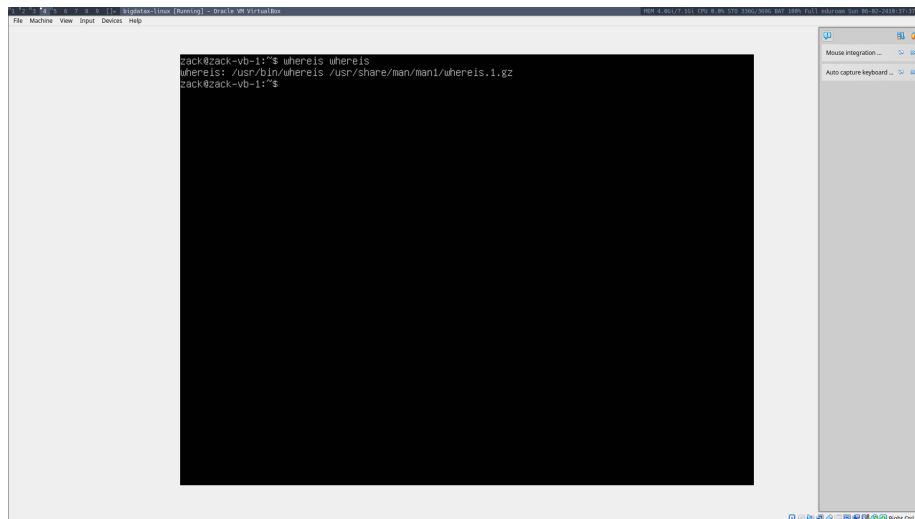
uname

The `uname` command is used to find information about the machine's operating system and hardware.



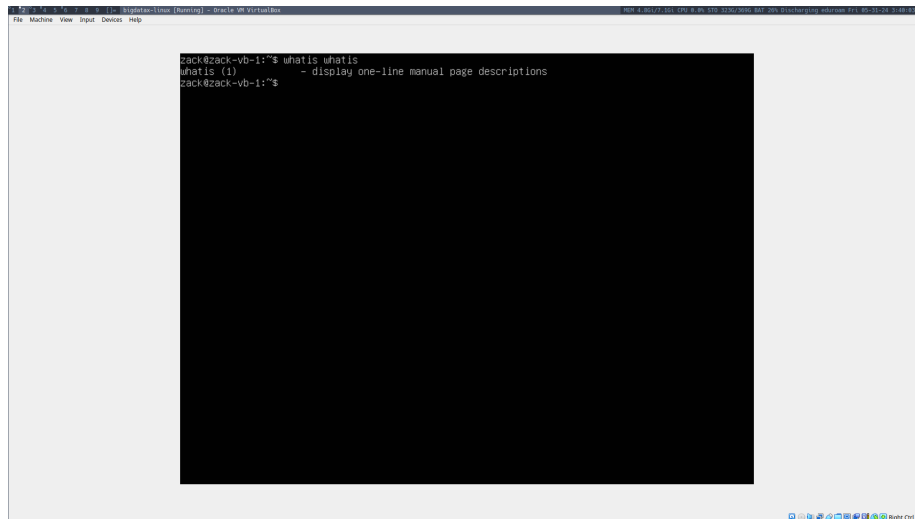
whereis

The **whereis** command is used to locate commands in the filesystem.



whatis

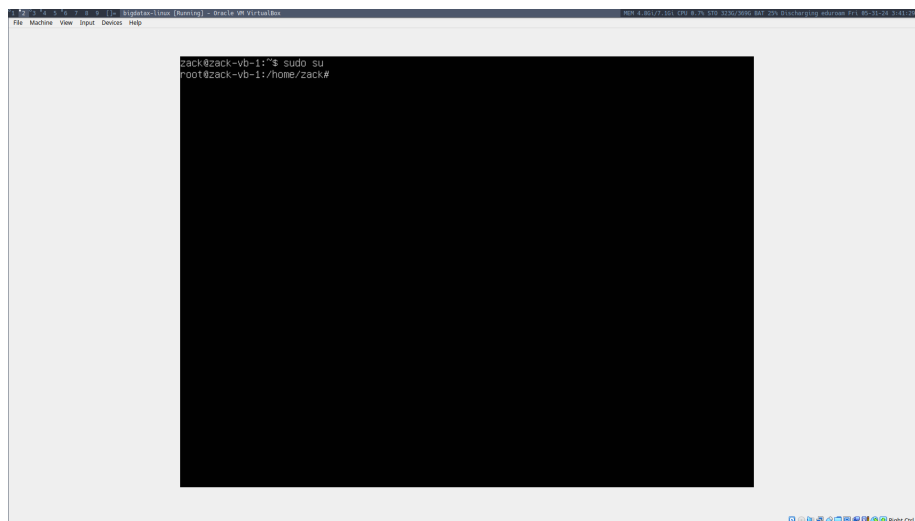
The **whatis** command displays one-line manual descriptions of commands.

A screenshot of a terminal window titled "Ubuntu [zack@zack:~]\$". The terminal shows the command `whatis` being entered, followed by its description: `- display one-line manual page descriptions`. The prompt `zack@zack-vb-1:~$` is visible at the bottom of the terminal area.

```
zack@zack-vb-1:~$ whatis
whatis (1) - display one-line manual page descriptions
zack@zack-vb-1:~$
```

su

The `su` command is used to change the active user, typically to root.

A screenshot of a terminal window titled "Ubuntu [zack@zack:~]\$". The terminal shows the command `sudo su` being entered, followed by the prompt `root@zack-vb-1:~#`. The prompt `zack@zack-vb-1:~$` is visible at the bottom of the terminal area.

```
zack@zack-vb-1:~$ sudo su
root@zack-vb-1:~#
```

ping

The `ping` command is used to check whether a host is reachable by sending it packets.

```
zack@zack-vb-1:~$ ping iit.edu
PING iit.edu (174.143.130.167) 56(84) bytes of data:
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=1 ttl=63 time=29.9 ms
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=2 ttl=63 time=28.6 ms
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=3 ttl=63 time=32.0 ms
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=4 ttl=63 time=31.3 ms
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=5 ttl=63 time=29.2 ms
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=6 ttl=63 time=29.9 ms
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=7 ttl=63 time=30.0 ms
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=8 ttl=63 time=29.6 ms
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=9 ttl=63 time=29.3 ms
64 bytes from www-c2.iit.edu (174.143.130.167): icmp_seq=10 ttl=63 time=30.5 ms
```

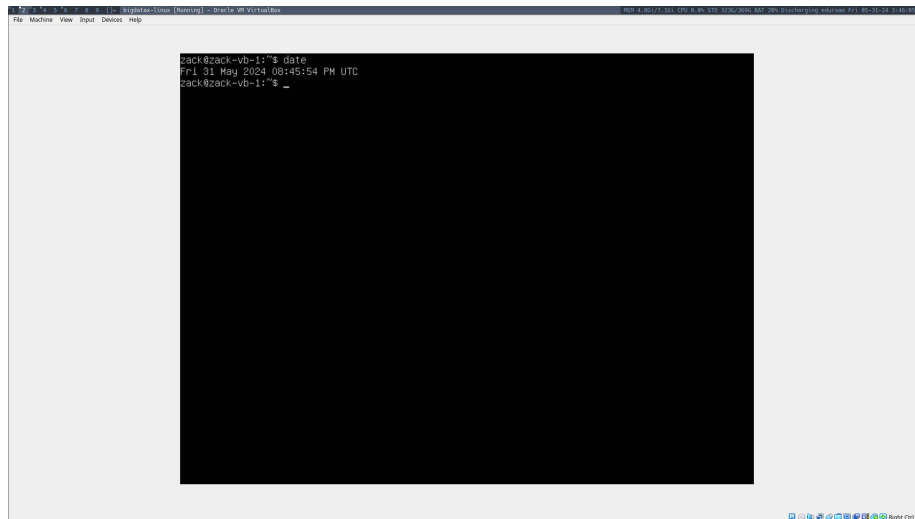
traceroute

The **traceroute** command is used to find the path on the network to a host.

```
zack@zack-vb-1:~$ traceroute iit.edu
traceroute to iit.edu (174.143.130.167), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.439 ms  0.353 ms  0.328 ms
 2  192.168.128.1 (192.168.128.1)  6.409 ms  6.515 ms  6.485 ms
 3  216.47.152.1 (216.47.152.1)  7.437 ms  7.180 ms  7.571 ms
 4  216.47.159.165 (216.47.159.165)  7.440 ms  7.493 ms  7.599 ms
 5  host-131-239-179-217.customer.veroxity.net (131.239.179.217)  8.062 ms  8.248 ms  8.222 ms
 6  ae30-chcg1ldfj91.lighttower.net (160.72.249.3)  9.961 ms  5.284 ms  5.237 ms
 7  144.121.109.123.lighttower.net (144.121.109.123)  5.093 ms  6.298 ms  6.432 ms
 8  * * *
 9  * * *
10  * * *
11  ae6.er3.dfw2.us.zip.zayo.com (64.125.26.55)  39.819 ms  39.880 ms  ae10.er3.dfw2.us.zip.zayo.com
   (64.125.26.63)  38.997 ms
12  64.125.46.126.IPVX-172935-900-2V0.zip.zayo.com (64.125.46.126)  39.966 ms  38.893 ms  37.630 ms
13  * * *
14  corec-dcpe4.dfw1.rackspace.net (148.62.41.103)  27.363 ms  corec-dcpe3.dfw1.rackspace.net (148.62
   .41.101)  66.988 ms  66.407 ms
15  core10-corec.dfw1.rackspace.net (148.62.41.195)  63.891 ms  core9-corec.dfw1.rackspace.net (148.6
   2.41.125)  75.741 ms  core10-corec.dfw1.rackspace.net (148.62.41.125)  26.113 ms
16  core10-aggr170a-7.dfw3.rackspace.net (67.192.88.89)  27.125 ms  core9-aggr170a-7.dfw3.rackspace.n
   et (67.192.88.111)  27.780 ms  core10-aggr170b-7.dfw3.rackspace.net (72.32.111.77)  37.346 ms
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * *
```

date

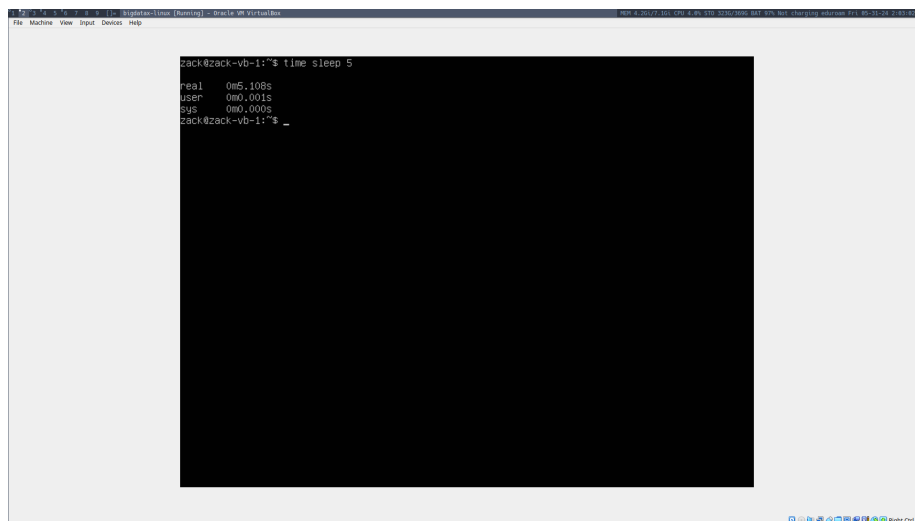
The **date** command prints the current date and time.



```
zack@zack-vb-1:~$ date
Fri 31 May 2024 08:45:54 PM UTC
zack@zack-vb-1:~$ _
```

time

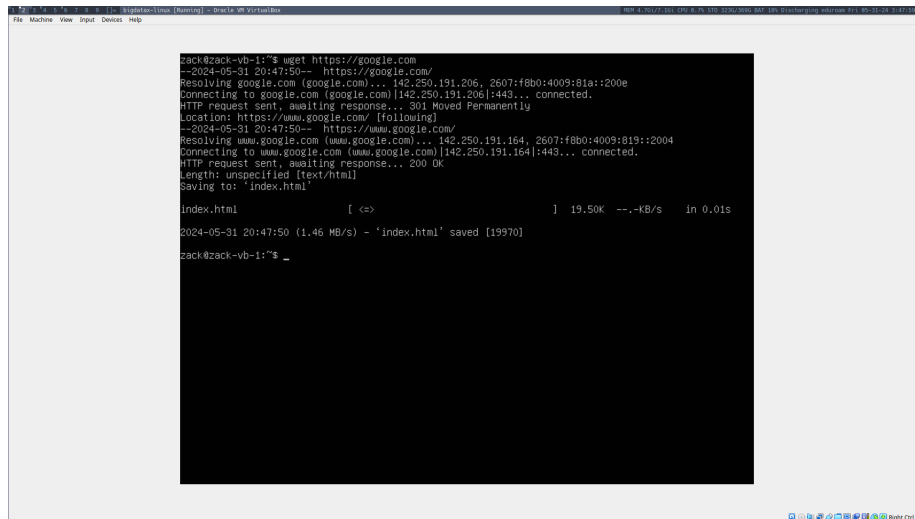
The **time** command measures the execution time of a given command.



```
zack@zack-vb-1:~$ time sleep 5
real    0m5.108s
user    0m0.001s
sys     0m0.000s
zack@zack-vb-1:~$ _
```

wget

The **wget** command downloads or uploads files on the network.

A screenshot of a terminal window titled "Ubuntu-20.04 [Running] - Oracle VM VirtualBox". The terminal shows a user named 'zack' at a prompt 'zack@zack-vb-1:~\$' running the command 'curl https://google.com'. The output shows the curl process resolving the domain, connecting to the server, and receiving a 301 Moved Permanently status. The user then runs 'curl https://www.google.com/' which returns a 200 OK status. The terminal also shows the file 'index.html' being saved to the disk.

```
zack@zack-vb-1:~$ curl https://google.com
--2024-05-31 20:47:50-- https://google.com/
Resolving google.com (google.com)... 142.250.191.206, 2607:f8b0:4009:81a::200e
Connecting to google.com (google.com)[142.250.191.206]:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.google.com/ [following]
--2024-05-31 20:47:50-- https://www.google.com/
Resolving www.google.com (www.google.com)... 142.250.191.164, 2607:f8b0:4009:819::2004
Connecting to www.google.com (www.google.com)[142.250.191.164]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

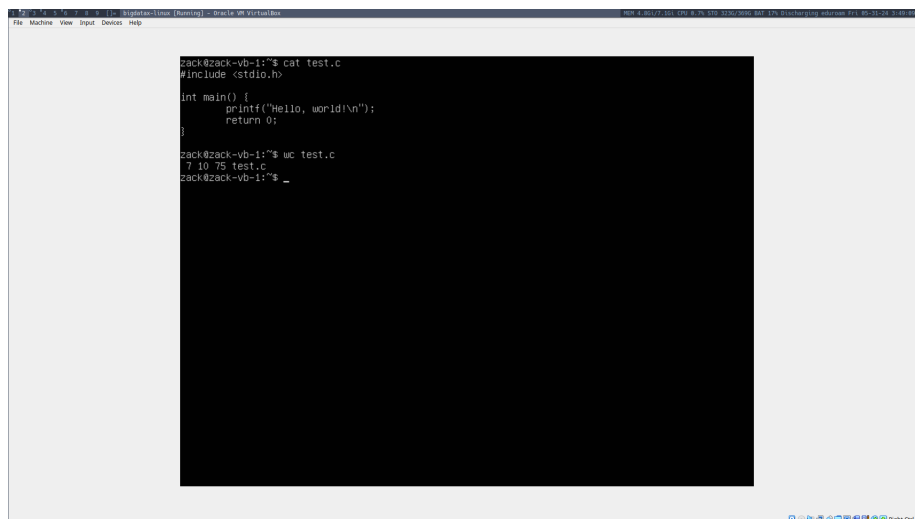
index.html           [ <> ] 19.50K --.-KB/s  in 0.01s

2024-05-31 20:47:50 (1.46 MB/s) - 'index.html' saved [19970]

zack@zack-vb-1:~$ _
```

wc

The **wc** (word count) command counts the lines, words, and bytes of files and streams.

A screenshot of a terminal window titled "Ubuntu-20.04 [Running] - Oracle VM VirtualBox". The terminal shows a user named 'zack' at a prompt 'zack@zack-vb-1:~\$' creating a file 'test.c' with the 'cat' command. The file contains a C program that prints 'Hello, world!'. The user then runs 'wc test.c' which outputs '7 10 75 test.c', indicating 7 lines, 10 words, and 75 bytes.

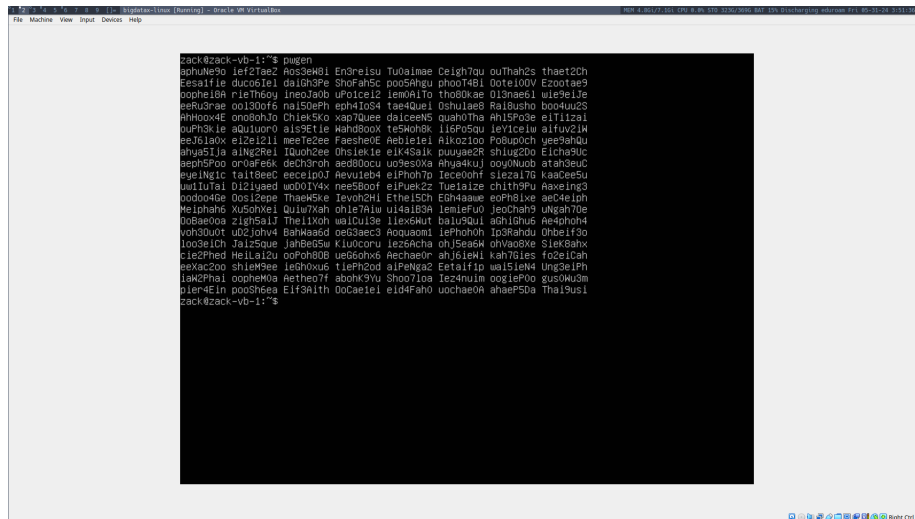
```
zack@zack-vb-1:~$ cat test.c
#include <stdio.h>

int main() {
    printf("Hello, world!\n");
    return 0;
}

zack@zack-vb-1:~$ wc test.c
7 10 75 test.c
zack@zack-vb-1:~$ _
```

pwgen

The **pwgen** command generates secure passwords that are meant to be easy to memorize.



III: Bash Scripting

The following bash script creates a data set of size \$2 named \$3 of integers and strings.

```
for i in $(seq 1 $2);
do
    echo $SRANDOM $SRANDOM abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
    qrstuvwxyabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz >> $3
done
```

The alphabet string is exactly 100 bytes long in ASCII because the script is encoded in UTF-8, a superset of ASCII. A->Y includes 25 characters, and so alphabet is repeated 3 more times to reach 100 total characters.

Next, we log the performance data of the bash script using the `time` command in the following file format:

```
N Create Sort
1000 0.0046 0.008
100000 1.519 0.105
10000000 140.64 13.903
```

The following Python script plots the data given by the `time.txt`.

```
from matplotlib import pyplot as plt

fig, ax = plt.subplots(figsize=(8, 12))
ax.set_title('Dataset Generation and Sorting Times vs. N')

file = open('time.txt', 'r')
```

```

file.readline() # Toss first line

N_data = []
create_data = []
sort_data = []

for i in range(3):
    parts = file.readline().split(' ')
    N_data.append(parts[0])
    create_data.append(parts[1])
    if parts[2][-1] == '\n':
        parts[2] = parts[2][:-1] # Trim newline if present
    sort_data.append(parts[2])

print(f"N: {N_data}, create: {create_data}, sort: {sort_data}")

ax.set_xlabel('N')
ax.set_ylabel('Time (s)')

# Draw lines
ax.plot(N_data, create_data, 'b')
ax.plot(N_data, sort_data, 'r')

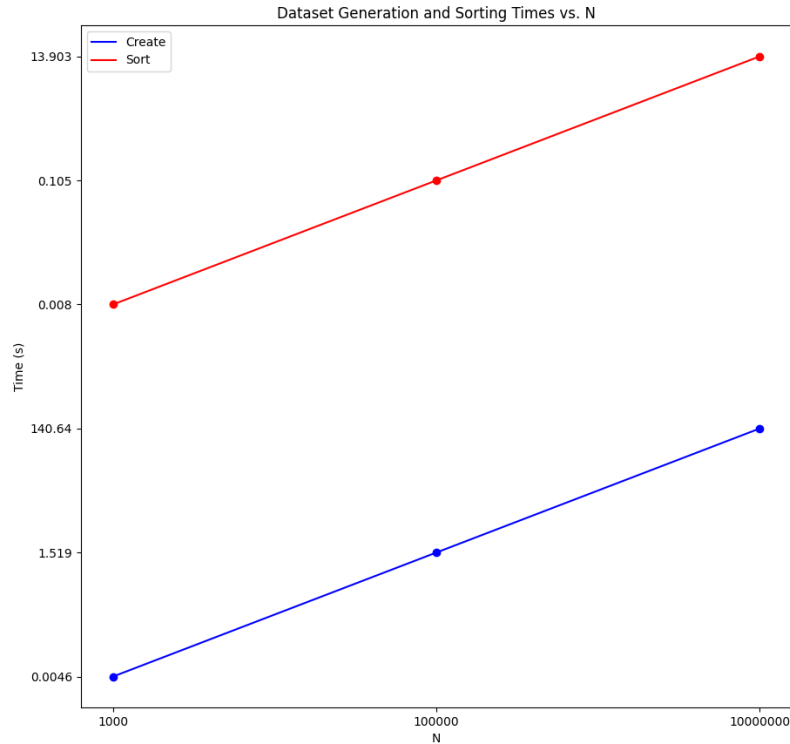
# Draw points
for i in range(len(N_data)):
    ax.scatter(N_data[i], create_data[i], color='b')
for i in range(len(N_data)):
    ax.scatter(N_data[i], sort_data[i], color='r')

plt.legend(['Create', 'Sort'])

plt.show()

```

Running the above Python script with the above data yields the following chart:



IV. Questions about VMs

1. In the system configuration of the VM, explain how changing the number of processors changes the behavior of your VM. Explain a scenario where you want to set this to the minimum, and a scenario where you want to set it to the maximum. Why is setting it to the maximum potentially a bad idea?

- The number of processors influences how quickly the VM can perform operations.
- You may want to set this parameter to the minimum possible value if you intend to create many VMs or need to run other intensive process on your host OS.
- You may want to set this parameter to the maximum if the VM will need to run compute-intensive programs. This is potentially a bad idea because it greatly reduces the amount of processors left for your host OS, which you need to use to manage the virtual machine and perform other tasks.

2. In the system configuration of the VM, under the Acceleration Tab, explain the difference between the paravirtualization options: None, Legacy, Minimal, Hyper-V, and KVM. Explain which one would be best to use with Ubuntu Linux, and why

- None: No paravirtualization is supported, meaning that the guest (VM) OS has no knowledge of the host OS and interfaces with emulated hardware.
- Legacy: Older paravirtualization interfaces are supported, which often lack the performance improvements found with modern paravirtualization techniques.
- Minimal: Some paravirtualization features are supported, granting some performance benefits.
- Hyper-V: Microsoft's Hyper-V hypervisor is supported, which provides good performance, but only for Windows host and guest operating systems.
- KVM: Kernel-based Virtual Machine (KVM) is supported, meaning that the Linux kernel functions as a hypervisor, managing the guest OS. Limitation: the host OS must be Linux-based.
- Verdict: KVM would be best with an Ubuntu host OS because it offers the best performance benefits using a technique that is compatible with the setup.

3. In storage devices when configuring the VM, there are multiple types of storage controllers: explain the difference between the IDE, SATA, and NVMe controller. Give an example for each type of storage controller of a scenario where you may want to use this type of controller.

- IDE: Integrated Drive Electronics is a dated standard for managing storage devices with relatively slow performance and transfer rates of up to 133 MB/s. It handles data transfers using a parallel interface. You may want to use an IDE controller for its compatibility with older systems and software.
- SATA: Serial Advanced Technology Attachment is a more modern standard that uses a serial interface to connect with storage devices. This could be a good option when high transfer speeds are not the most critical priority, and when a balance of speed and compatibility is necessary.
- NVMe: Non-Volatile Memory Express is a standard meant solely for SSDs using the PCI express bus on the motherboard. It offers significantly higher performance than SATA. A potential use case would be an application that requires many high-speed storage calls, like a database server for web applications.

4. In the network configuration of the VM, there are multiple types of network adapters: explain the difference between NAT, Bridged

Adapter, Internal Network, and Host-only Network. Give an example for each type of network of a scenario where you may want to use this type of network.

- NAT: Network Address Translation allows the virtual machine to share the host machine's IP address. The VM can access the internet and other networks, but other devices cannot start communication with the VM itself. You may want to use this type of network when creating a VM that requires internet access to run scripts.
- Bridged Adapter: This configuration connects the VM directly to the physical network, behaving like a separate machine on the network and possessing an independent IP address, which can be accessed by other network devices. This could be used to use the VM as a web server for the local network.
- Internal Network: This configuration connects the VMs on an network in which they can communicate with each other, but they are unable to communicate with the host and any external networks. This could be useful for creating an isolated networking testbed between several VMs.
- Host-only network: This setup is an internal network with a connection to the host machine. This could be useful for running a local web server that requires a Linux-based OS and accessing the website on the host's browser.

5. For the USB configuration of the VM, explain the difference between USB 1.1, 2.0, and 3.0 controllers.

- USB 1.1: This is an older USB standard with a maximum transfer rate of 12 Mbps. This could be useful for using legacy software and hardware with the VM. However, 12 Mbps is a major limitation for many applications.
- USB 2.0: This is a popular standard with a maximum transfer rate of 480 Mbps. This is widely compatible and fast, which makes it a good option for applications that don't require blazing transmission speeds.
- USB 3.0: This is a newer standard with a maximum transfer rate of 5 Gbps, making it very fast. Additionally, it is backwards compatible with USB 2.0 and 1.1 interfaces. This means that USB 3.0 is a great choice for fast and compatible data transmission.