Homework 4

FROM: Jonas Zhonghan Xie
TO: Raj
SUBJECT: RE: Questions about Database

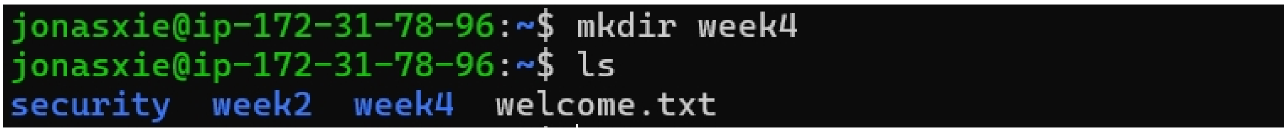
Hi Raj,

Thanks for reaching out! I am happy to help with your questions about the text files in the database.

Part 1

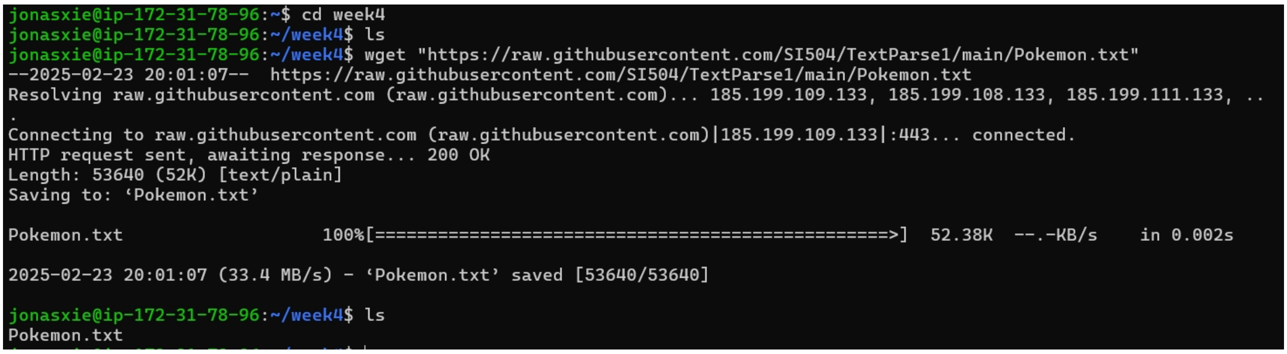
1. Our database servers are hosted on Michigan Academic Computing Center (MACC) which is a large 2 MW data center located in Ann Arbor. The data center is equipped with backup power supplies, very reliable cooling systems (AC cooling aisles for example) and high-speed network connections. With the uninterrupted power supply (UPS) and the backup generators, the data center can be still running even in case of power outages. So don't worry about it. Our data will be safe. And the cooling systems are also very reliable. There are many cooling aisles, and the temperature is kept at a pretty constant level. The center is also equipped with well-designed fire suppression systems.

2. I used `mkdir` to create a folder called `week4` in the home directory.



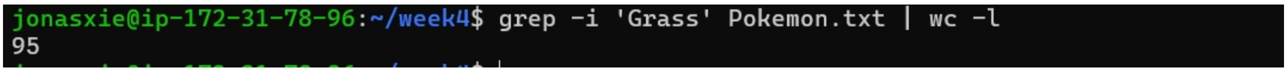
```
jonasxie@ip-172-31-78-96:~$ mkdir week4
jonasxie@ip-172-31-78-96:~$ ls
security  week2  week4  welcome.txt
```

3. I used `wget` to download the `pokemon.txt` in the directory.



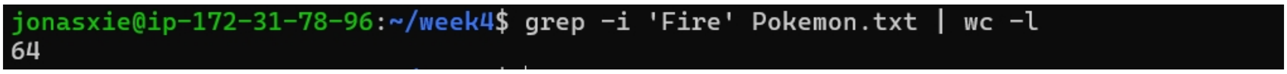
```
jonasxie@ip-172-31-78-96:~$ cd week4
jonasxie@ip-172-31-78-96:~/week4$ ls
jonasxie@ip-172-31-78-96:~/week4$ wget "https://raw.githubusercontent.com/SI504/TextParse1/main/Pokemon.txt"
--2025-02-23 20:01:07--  https://raw.githubusercontent.com/SI504/TextParse1/main/Pokemon.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.111.133, ..
.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 53640 (52K) [text/plain]
Saving to: ‘Pokemon.txt’

Pokemon.txt                   100%[================================================>]  52.38K  --.-KB/s    in 0.002s

2025-02-23 20:01:07 (33.4 MB/s) - ‘Pokemon.txt’ saved [53640/53640]

jonasxie@ip-172-31-78-96:~/week4$ ls
Pokemon.txt
```

4. I used `grep -i` to count the occurrences of `Grass` and passed it to `wc -l` to count the lines. There are 95 'Grass Pokemons' in the file.



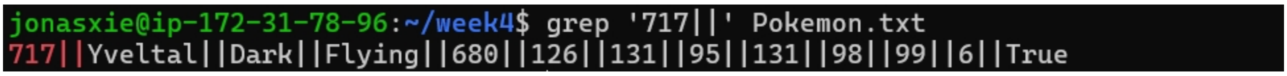
```
jonasxie@ip-172-31-78-96:~/week4$ grep -i 'Grass' Pokemon.txt | wc -l
95
```

5. I used the same command to count the occurrences of `Fire` and passed it to `wc -l` to count the lines. There are 64 'Fire Pokemons' in the file.



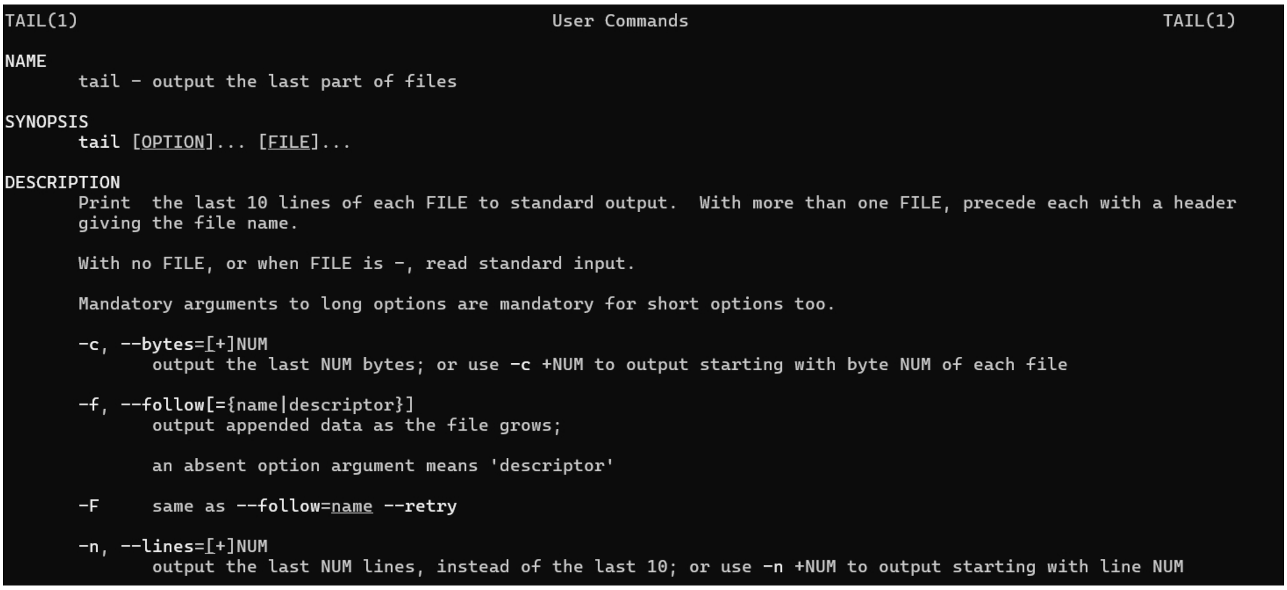
```
jonasxie@ip-172-31-78-96:~/week4$ grep -i 'Fire' Pokemon.txt | wc -l
64
```

6. I used `grep '711||` to locate the 711th pokemon in the file. It is Yveltal, a dark flying pokemon.



```
jonasxie@ip-172-31-78-96:~/week4$ grep '717||' Pokemon.txt
717||Yveltal||Dark||Flying||680||126||131||95||131||98||99||6||True
```

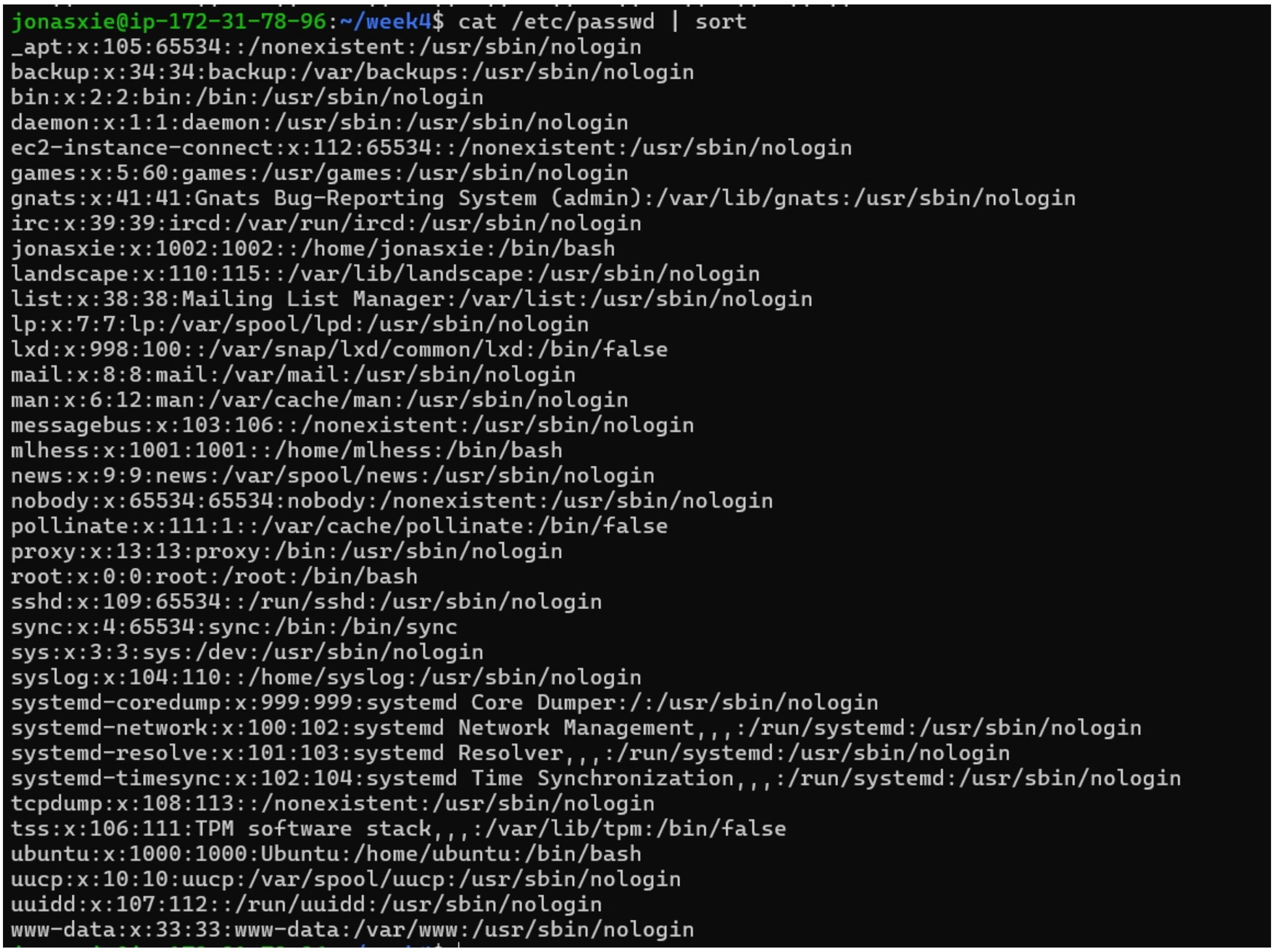
7. I used `tail -n 5` command to show the last 5 lines in the pokemon file. The last five pokemon are "Dianche", "DiancieMega Diancie", "HoopaHoopa Confined", "HoopaHoopa Unbound", and "Volcanion".



```
TAIL(1)                              User Commands                              TAIL(1)

NAME
       tail - output the last part of files

SYNOPSIS
       tail [OPTION]... [FILE]...

DESCRIPTION
       Print  the last 10 lines of each FILE to standard output.  With more than one FILE, precede each with a header
       giving the file name.

       With no FILE, or when FILE is -, read standard input.

       Mandatory arguments to long options are mandatory for short options too.

       -c, --bytes=[+]NUM
              output the last NUM bytes; or use -c +NUM to output starting with byte NUM of each file

       -f, --follow[={name|descriptor}]
              output appended data as the file grows;

              an absent option argument means 'descriptor'

       -F     same as --follow=name --retry

       -n, --lines=[+]NUM
              output the last NUM lines, instead of the last 10; or use -n +NUM to output starting with line NUM
```

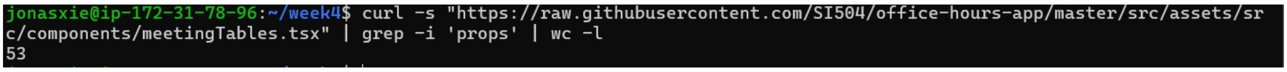
Part 2

1. You can use `cat /etc/passwd | sort` to show the users in alphabetical order.



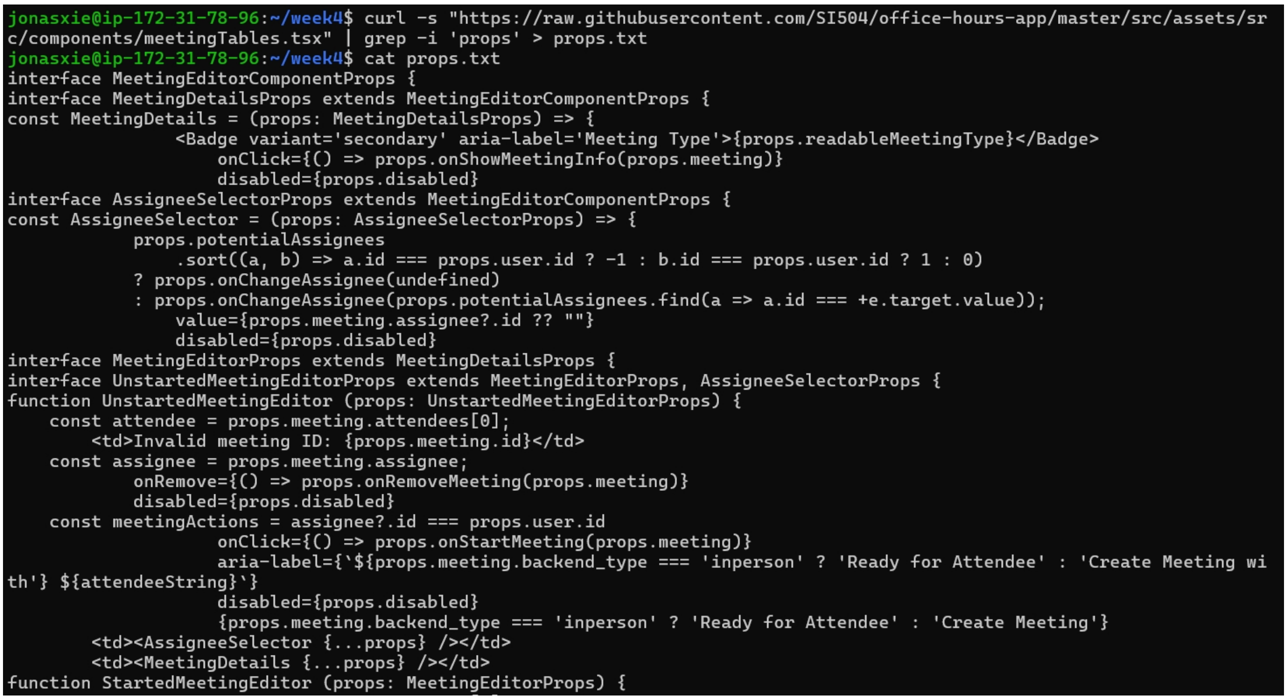
```
jonasxie@ip-172-31-78-96:~/week4$ cat /etc/passwd | sort
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
ec2-instance-connect:x:112:65534::/nonexistent:/usr/sbin/nologin
games:x:5:60:games:/usr/games:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
jonasxie:x:1002:1002::/home/jonasxie:/bin/bash
landscape:x:110:115::/var/lib/landscape:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
mlhess:x:1001:1001::/home/mlhess:/bin/bash
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
pollinate:x:111:1::/var/cache/pollinate:/bin/false
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
root:x:0:0:root:/root:/bin/bash
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
sys:x:3:3:sys:/dev:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

2. I used `curl -s` to fetch the file and then used `grep` to find the lines that contain the word "props" and passed it to `wc -l` to count the lines. There are 53 lines containing the word, case insensitive.



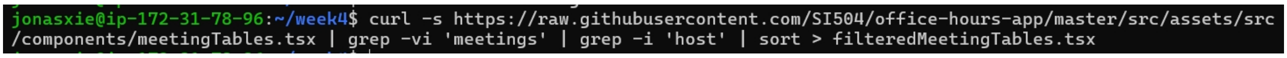
```
jonasxie@ip-172-31-78-96:~/week4$ curl -s "https://raw.githubusercontent.com/SI504/office-hours-app/master/src/assets/sr
c/components/meetingTables.tsx" | grep -i 'props' | wc -l
53
```

3. I used the same command to download, find the words. And then I passed the output to 'props.txt" using `> props.txt` command. The file is created in the directory.

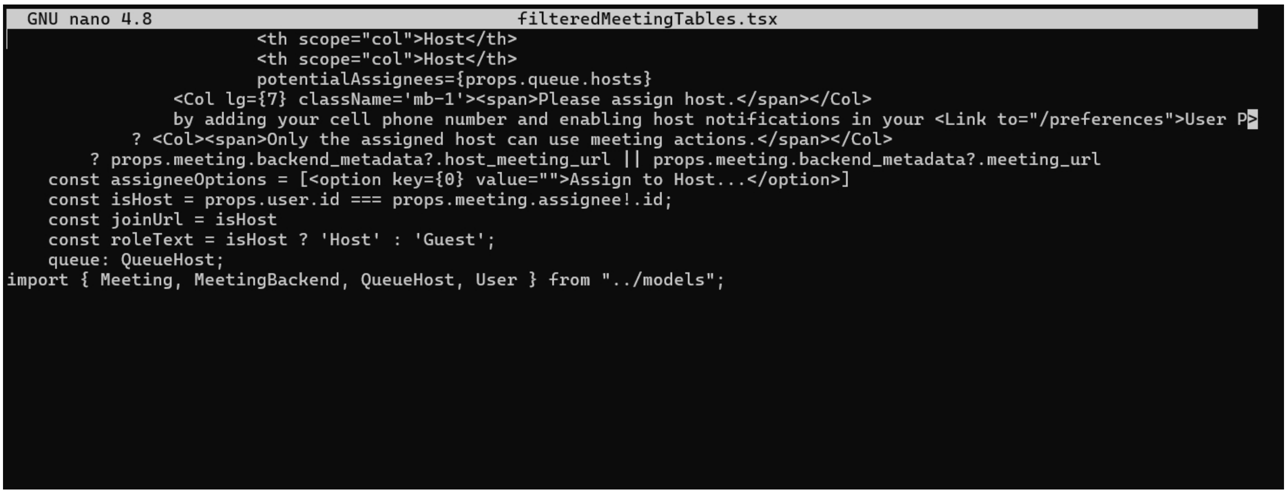


```
jonasxie@ip-172-31-78-96:~/week4$ curl -s "https://raw.githubusercontent.com/SI504/office-hours-app/master/src/assets/sr
c/components/meetingTables.tsx" | grep -i 'props' > props.txt
jonasxie@ip-172-31-78-96:~/week4$ cat props.txt
interface MeetingEditorComponentProps {
interface MeetingDetailsProps extends MeetingEditorComponentProps {
const MeetingDetails = (props: MeetingDetailsProps) => {
                <Badge variant='secondary' aria-label='Meeting Type'>{props.readableMeetingType}</Badge>
                    onClick={() => props.onShowMeetingInfo(props.meeting)}
                    disabled={props.disabled}
interface AssigneeSelectorProps extends MeetingEditorComponentProps {
const AssigneeSelector = (props: AssigneeSelectorProps) => {
            props.potentialAssignees
                .sort((a, b) => a.id === props.user.id ? -1 : b.id === props.user.id ? 1 : 0)
            ? props.onChangeAssignee(undefined)
            : props.onChangeAssignee(props.potentialAssignees.find(a => a.id === +e.target.value));
                value={props.meeting.assignee?.id ?? ""}
                disabled={props.disabled}
interface MeetingEditorProps extends MeetingDetailsProps {
interface UnstartedMeetingEditorProps extends MeetingEditorProps, AssigneeSelectorProps {
function UnstartedMeetingEditor (props: UnstartedMeetingEditorProps) {
    const attendee = props.meeting.attendees[0];
        <td>Invalid meeting ID: {props.meeting.id}</td>
    const assignee = props.meeting.assignee;
            onRemove={() => props.onRemoveMeeting(props.meeting)}
            disabled={props.disabled}
    const meetingActions = assignee?.id === props.user.id
                    onClick={() => props.onStartMeeting(props.meeting)}
                    aria-label={`${props.meeting.backend_type === 'inperson' ? 'Ready for Attendee' : 'Create Meeting wi
th'} ${attendeeString}`}
                    disabled={props.disabled}
                    {props.meeting.backend_type === 'inperson' ? 'Ready for Attendee' : 'Create Meeting'}
        <td><AssigneeSelector {...props} /></td>
        <td><MeetingDetails {...props} /></td>
function StartedMeetingEditor (props: MeetingEditorProps) {
```

4. I first used `curl` to download the file and then used `grep -vi` to find the lines without "meetings", then passed it to `grep -i` to find the lines with "host", then passed it to `sort` and finally saved it to the file.



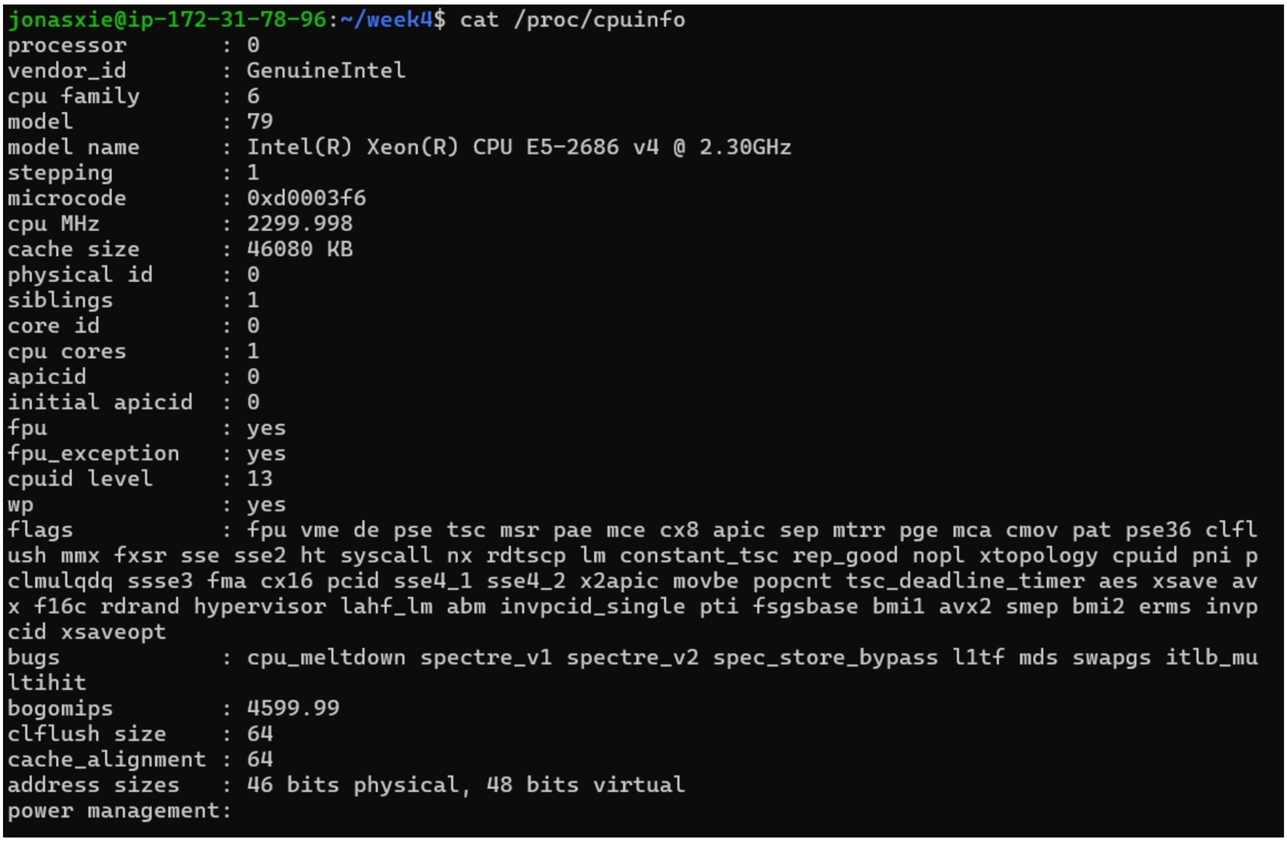
```
jonasxie@ip-172-31-78-96:~/week4$ curl -s https://raw.githubusercontent.com/SI504/office-hours-app/master/src/assets/src
/components/meetingTables.tsx | grep -vi 'meetings' | grep -i 'host' | sort > filteredMeetingTables.tsx
```



```
  GNU nano 4.8                                   filteredMeetingTables.tsx
                        <th scope="col">Host</th>
                        <th scope="col">Host</th>
                        potentialAssignees={props.queue.hosts}
                <Col lg={7} className='mb-1'><span>Please assign host.</span></Col>
                by adding your cell phone number and enabling host notifications in your <Link to="/preferences">User P>
            ? <Col><span>Only the assigned host can use meeting actions.</span></Col>
        ? props.meeting.backend_metadata?.host_meeting_url || props.meeting.backend_metadata?.meeting_url
    const assigneeOptions = [<option key={0} value="">Assign to Host...</option>]
    const isHost = props.user.id === props.meeting.assignee!.id;
    const joinUrl = isHost
    const roleText = isHost ? 'Host' : 'Guest';
    queue: QueueHost;
import { Meeting, MeetingBackend, QueueHost, User } from "../models";
```

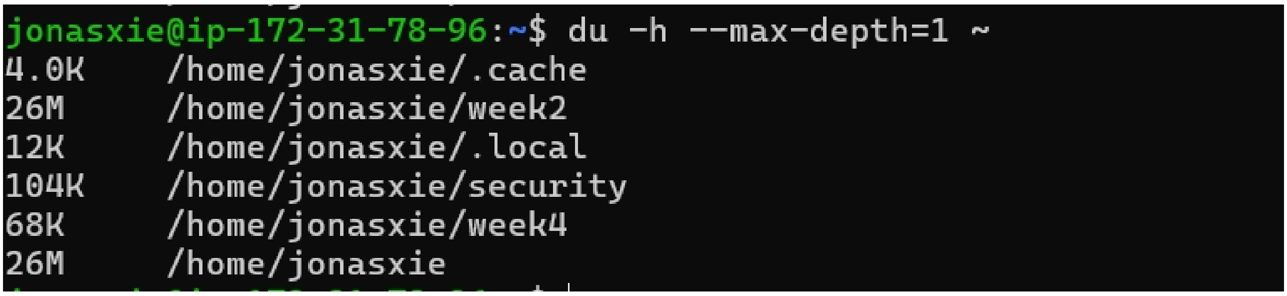
Part 3:

1. There is only one CPU on my server.



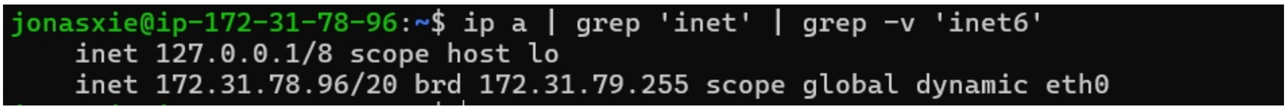
```
jonasxie@ip-172-31-78-96:~/week4$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 79
model name      : Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
stepping        : 1
microcode       : 0xd0003f6
cpu MHz         : 2299.998
cache size      : 46080 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clfl
ush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology cpuid pni p
clmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave av
x f16c rdrand hypervisor lahf_lm abm invpcid_single pti fsgsbase bmi1 avx2 smep bmi2 erms invp
cid xsaveopt
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swapgs itlb_mu
ltihit
bogomips        : 4599.99
clflush size    : 64
cache_alignment : 64
address sizes   : 46 bits physical, 48 bits virtual
power management:
```

2. I used `du` command to show the disk usage of the current directory. My home directory takes up 26M space in total. The subdirectories (`--max-depth=1`) are also shown in the output.



```
jonasxie@ip-172-31-78-96:~$ du -h --max-depth=1 ~
4.0K    /home/jonasxie/.cache
26M     /home/jonasxie/week2
12K     /home/jonasxie/.local
104K    /home/jonasxie/security
68K     /home/jonasxie/week4
26M     /home/jonasxie
```

3. I used `ip a` and `grep` to locate the IP address and filtered out `inet6`.



```
jonasxie@ip-172-31-78-96:~$ ip a | grep 'inet' | grep -v 'inet6'
    inet 127.0.0.1/8 scope host lo
    inet 172.31.78.96/20 brd 172.31.79.255 scope global dynamic eth0
```

Please let me know if you have any further questions. I am happy to help.

Best,
Jonas