Andree Salvo

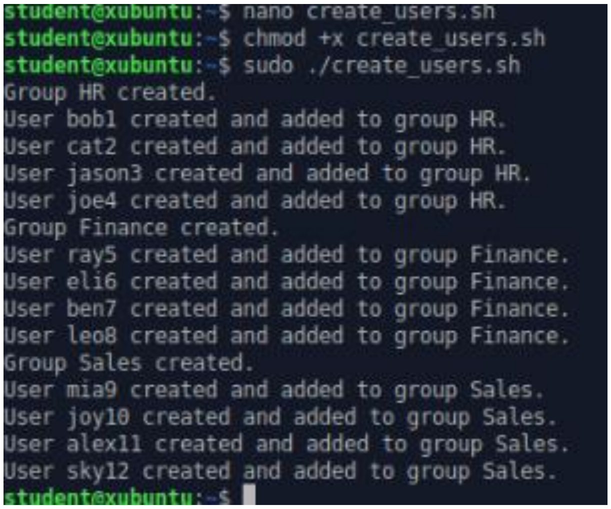
Southern New Hampshire University

CYB 300

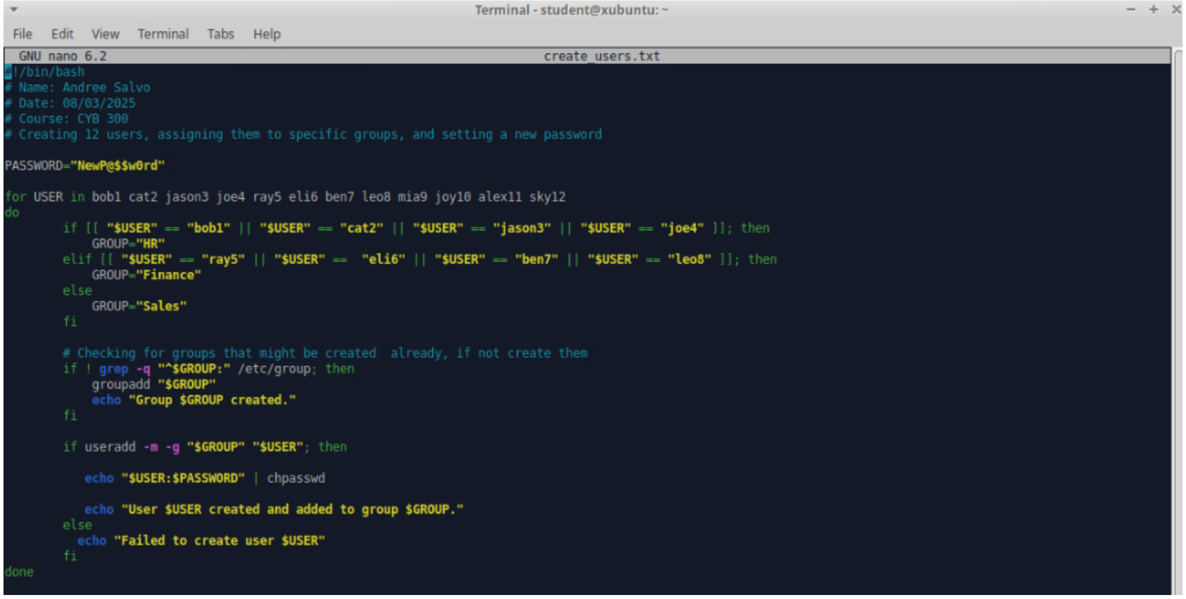
6-3 Activity

Instructor: Jason Keltner

Create three groups: Human Resources, Finance, and Sales. Create 12 user accounts and place them in one of the three groups. Set the passwords to NewP@$$w0rd



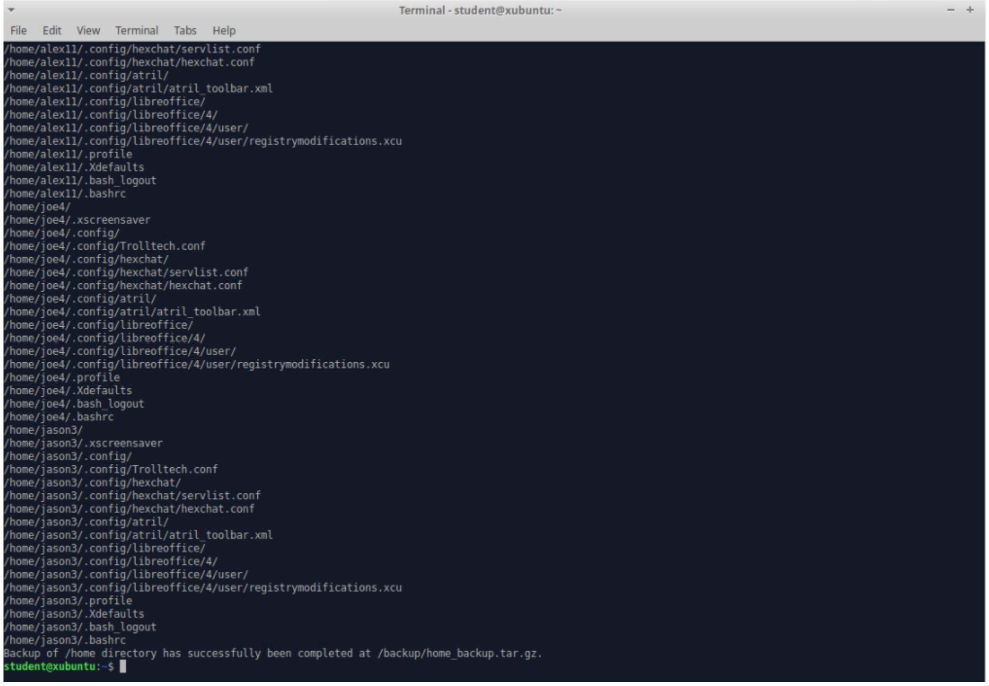
```
student@xubuntu:~$ nano create_users.sh
student@xubuntu:~$ chmod +x create_users.sh
student@xubuntu:~$ sudo ./create_users.sh
Group HR created.
User bob1 created and added to group HR.
User cat2 created and added to group HR.
User jason3 created and added to group HR.
User joe4 created and added to group HR.
Group Finance created.
User ray5 created and added to group Finance.
User eli6 created and added to group Finance.
User ben7 created and added to group Finance.
User leo8 created and added to group Finance.
Group Sales created.
User mia9 created and added to group Sales.
User joy10 created and added to group Sales.
User alex11 created and added to group Sales.
User sky12 created and added to group Sales.
student@xubuntu:~$
```



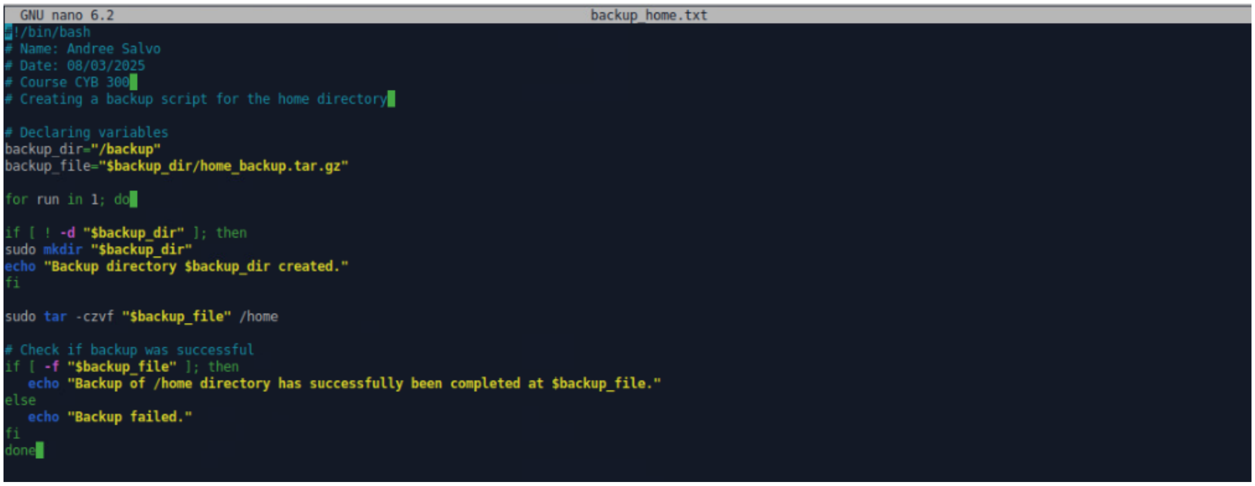
Terminal - student@xubuntu: ~
File Edit View Terminal Tabs Help

```
  GNU nano 6.2                          create_users.txt
#!/bin/bash
# Name: Andree Salvo
# Date: 08/03/2025
# Course: CYB 300
# Creating 12 users, assigning them to specific groups, and setting a new password

PASSWORD="NewP@$$w0rd"

for USER in bob1 cat2 jason3 joe4 ray5 eli6 ben7 leo8 mia9 joy10 alex11 sky12
do
        if [[ "$USER" == "bob1" || "$USER" == "cat2" || "$USER" == "jason3" || "$USER" == "joe4" ]]; then
            GROUP="HR"
        elif [[ "$USER" == "ray5" || "$USER" ==  "eli6" || "$USER" == "ben7" || "$USER" == "leo8" ]]; then
            GROUP="Finance"
        else
            GROUP="Sales"
        fi

        # Checking for groups that might be created  already, if not create them
        if ! grep -q "^$GROUP:" /etc/group; then
            groupadd "$GROUP"
            echo "Group $GROUP created."
        fi

        if useradd -m -g "$GROUP" "$USER"; then

           echo "$USER:$PASSWORD" | chpasswd

           echo "User $USER created and added to group $GROUP."
        else
          echo "Failed to create user $USER"
        fi
done
```

Create a script to back up the /home directory as a compressed tar file to a folder called /backup

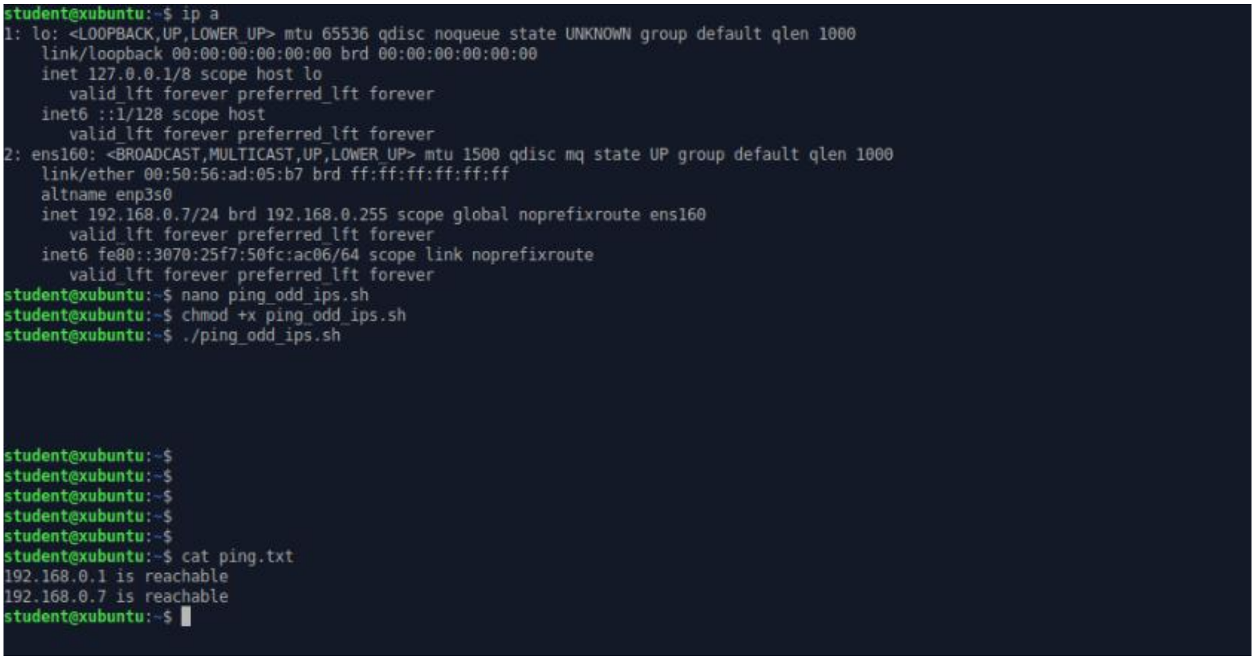

Terminal - student@xubuntu: ~
File Edit View Terminal Tabs Help

```
/home/alex11/.config/hexchat/servlist.conf
/home/alex11/.config/hexchat/hexchat.conf
/home/alex11/.config/atril/
/home/alex11/.config/atril/atril_toolbar.xml
/home/alex11/.config/libreoffice/
/home/alex11/.config/libreoffice/4/
/home/alex11/.config/libreoffice/4/user/
/home/alex11/.config/libreoffice/4/user/registrymodifications.xcu
/home/alex11/.profile
/home/alex11/.Xdefaults
/home/alex11/.bash_logout
/home/alex11/.bashrc
/home/joe4/
/home/joe4/.xscreensaver
/home/joe4/.config/
/home/joe4/.config/Trolltech.conf
/home/joe4/.config/hexchat/
/home/joe4/.config/hexchat/servlist.conf
/home/joe4/.config/hexchat/hexchat.conf
/home/joe4/.config/atril/
/home/joe4/.config/atril/atril_toolbar.xml
/home/joe4/.config/libreoffice/
/home/joe4/.config/libreoffice/4/
/home/joe4/.config/libreoffice/4/user/
/home/joe4/.config/libreoffice/4/user/registrymodifications.xcu
/home/joe4/.profile
/home/joe4/.Xdefaults
/home/joe4/.bash_logout
/home/joe4/.bashrc
/home/jason3/
/home/jason3/.xscreensaver
/home/jason3/.config/
/home/jason3/.config/Trolltech.conf
/home/jason3/.config/hexchat/
/home/jason3/.config/hexchat/servlist.conf
/home/jason3/.config/hexchat/hexchat.conf
/home/jason3/.config/atril/
/home/jason3/.config/atril/atril_toolbar.xml
/home/jason3/.config/libreoffice/
/home/jason3/.config/libreoffice/4/
/home/jason3/.config/libreoffice/4/user/
/home/jason3/.config/libreoffice/4/user/registrymodifications.xcu
/home/jason3/.profile
/home/jason3/.Xdefaults
/home/jason3/.bash_logout
/home/jason3/.bashrc
Backup of /home directory has successfully been completed at /backup/home_backup.tar.gz.
student@xubuntu:~$
```

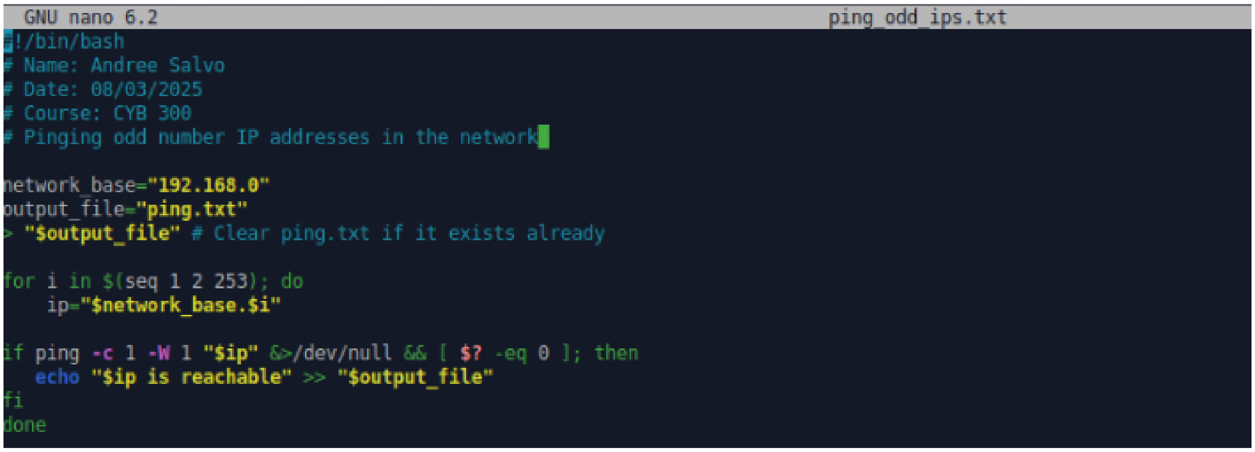


```
#!/bin/bash
# Name: Andree Salvo
# Date: 08/03/2025
# Course CYB 300
# Creating a backup script for the home directory

# Declaring variables
backup_dir="/backup"
backup_file="$backup_dir/home_backup.tar.gz"

for run in 1; do

if [ ! -d "$backup_dir" ]; then
sudo mkdir "$backup_dir"
echo "Backup directory $backup_dir created."
fi

sudo tar -czvf "$backup_file" /home

# Check if backup was successful
if [ -f "$backup_file" ]; then
   echo "Backup of /home directory has successfully been completed at $backup_file."
else
   echo "Backup failed."
fi
done
```

Use the Ping utility to report connections of all IP addresses that end in an odd number in the network, and output them to a text file called ping.txt



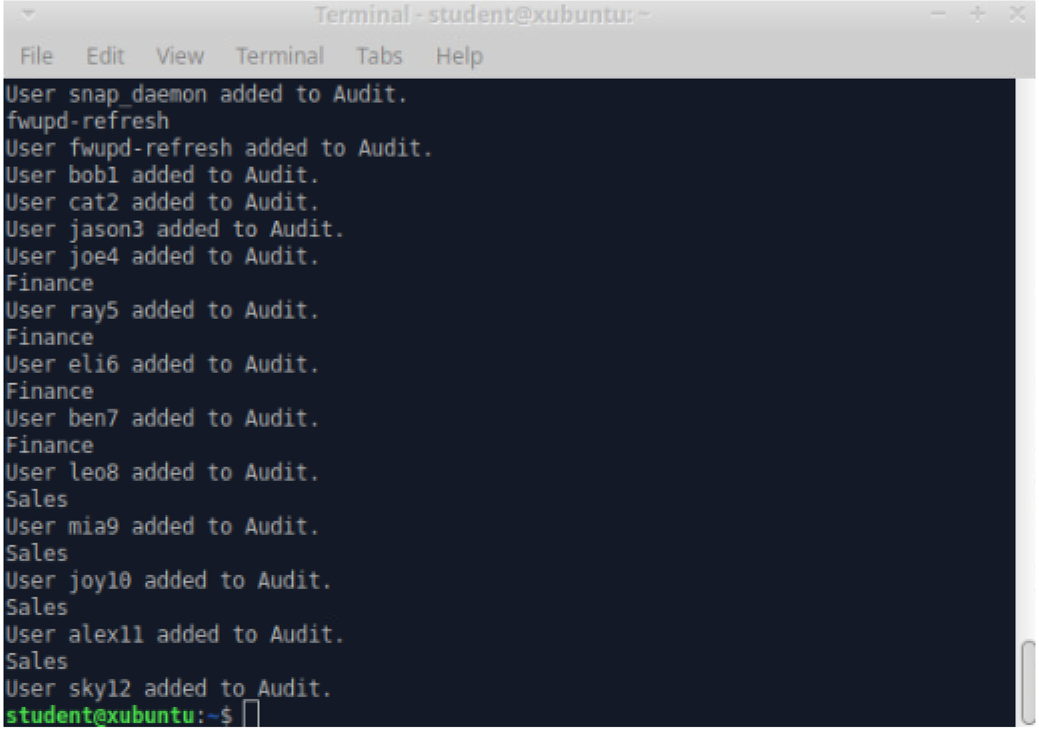
```
student@xubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:ad:05:b7 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.0.7/24 brd 192.168.0.255 scope global noprefixroute ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::3070:25f7:50fc:ac06/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
student@xubuntu:~$ nano ping_odd_ips.sh
student@xubuntu:~$ chmod +x ping_odd_ips.sh
student@xubuntu:~$ ./ping_odd_ips.sh




student@xubuntu:~$
student@xubuntu:~$
student@xubuntu:~$
student@xubuntu:~$
student@xubuntu:~$
student@xubuntu:~$ cat ping.txt
192.168.0.1 is reachable
192.168.0.7 is reachable
student@xubuntu:~$
```



```
  GNU nano 6.2                                                    ping_odd_ips.txt
#!/bin/bash
# Name: Andree Salvo
# Date: 08/03/2025
# Course: CYB 300
# Pinging odd number IP addresses in the network

network_base="192.168.0"
output_file="ping.txt"
> "$output_file" # Clear ping.txt if it exists already

for i in $(seq 1 2 253); do
    ip="$network_base.$i"

if ping -c 1 -W 1 "$ip" &>/dev/null && [ $? -eq 0 ]; then
   echo "$ip is reachable" >> "$output_file"
fi
done
```

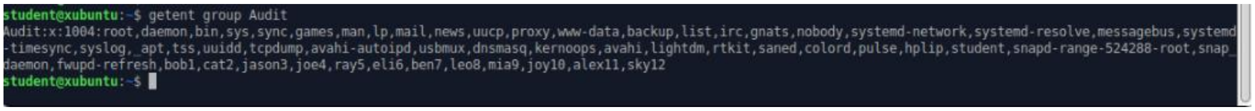
Create a user group named Audit and include all the employees of the HR and Finance organizational units

Note: I mistakenly misspelled “audit” as “Aduit”



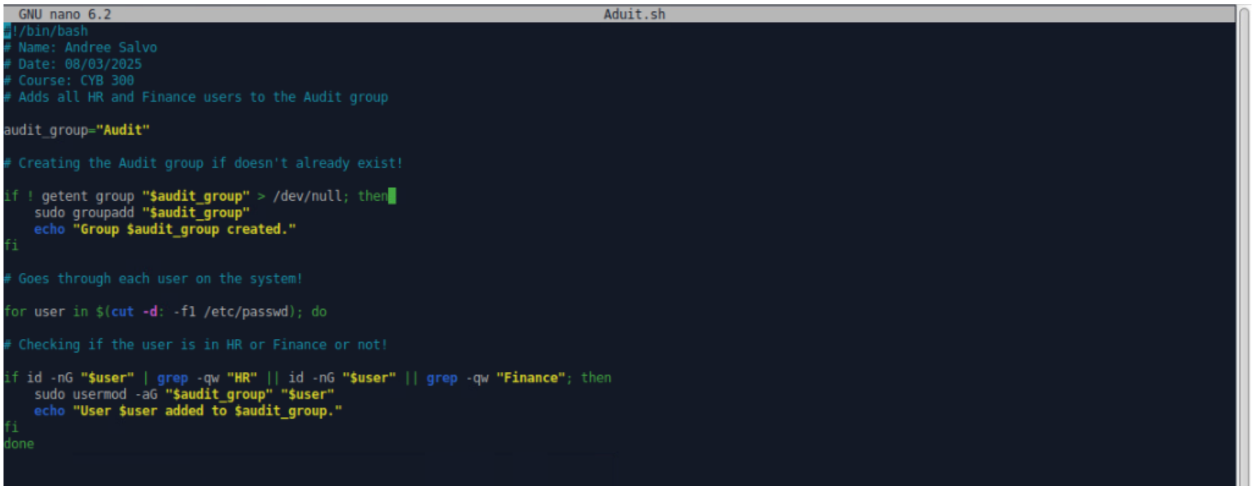
```
User snap_daemon added to Audit.
fwupd-refresh
User fwupd-refresh added to Audit.
User bob1 added to Audit.
User cat2 added to Audit.
User jason3 added to Audit.
User joe4 added to Audit.
Finance
User ray5 added to Audit.
Finance
User eli6 added to Audit.
Finance
User ben7 added to Audit.
Finance
User leo8 added to Audit.
Sales
User mia9 added to Audit.
Sales
User joy10 added to Audit.
Sales
User alex11 added to Audit.
Sales
User sky12 added to Audit.
student@xubuntu:~$
```

Results



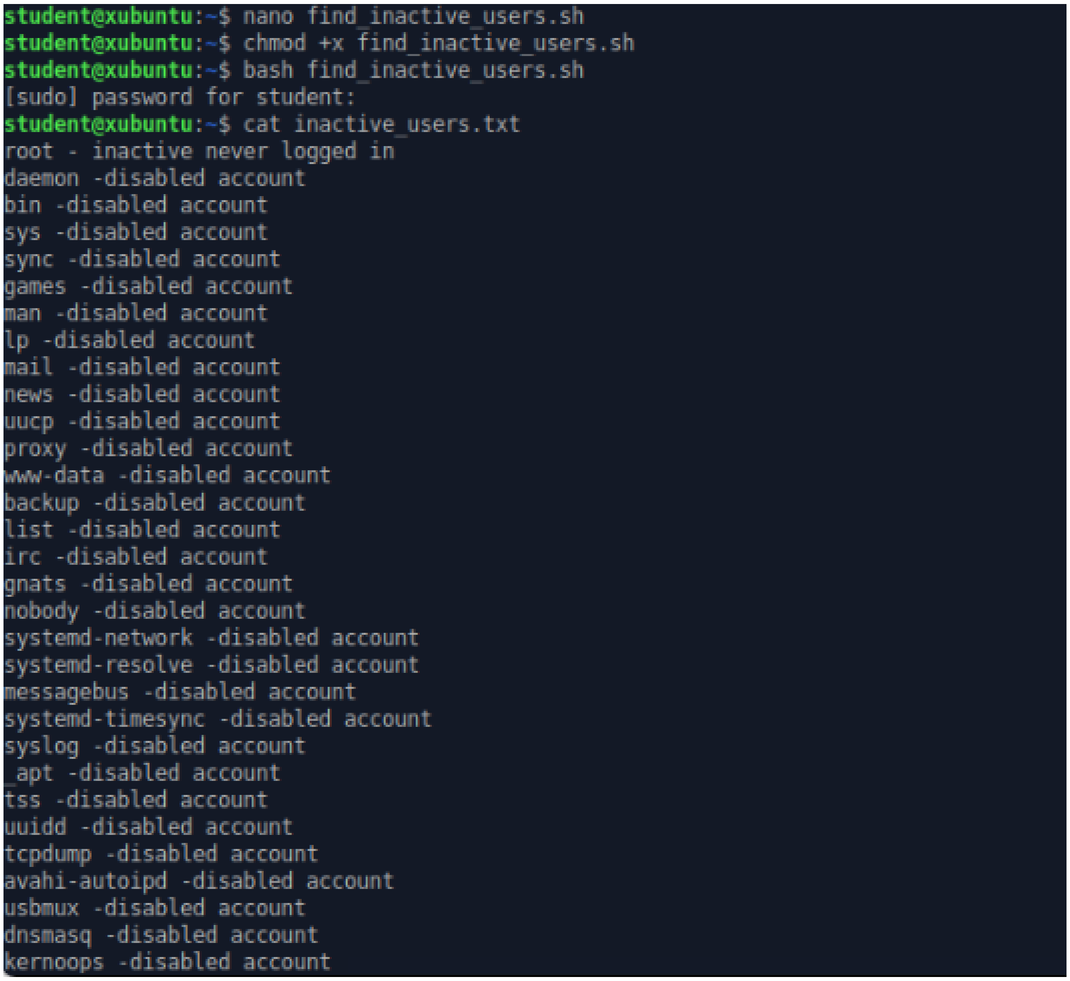
```
student@xubuntu:~$ getent group Audit
Audit:x:1004:root,daemon,bin,sys,sync,games,man,lp,mail,news,uucp,proxy,www-data,backup,list,irc,gnats,nobody,systemd-network,systemd-resolve,messagebus,systemd
-timesync,syslog,_apt,tss,uuidd,tcpdump,avahi-autoipd,usbmux,dnsmasq,kernoops,avahi,lightdm,rtkit,saned,colord,pulse,hplip,student,snapd-range-524288-root,snap_
daemon,fwupd-refresh,bob1,cat2,jason3,joe4,ray5,eli6,ben7,leo8,mia9,joy10,alex11,sky12
student@xubuntu:~$
```

Code



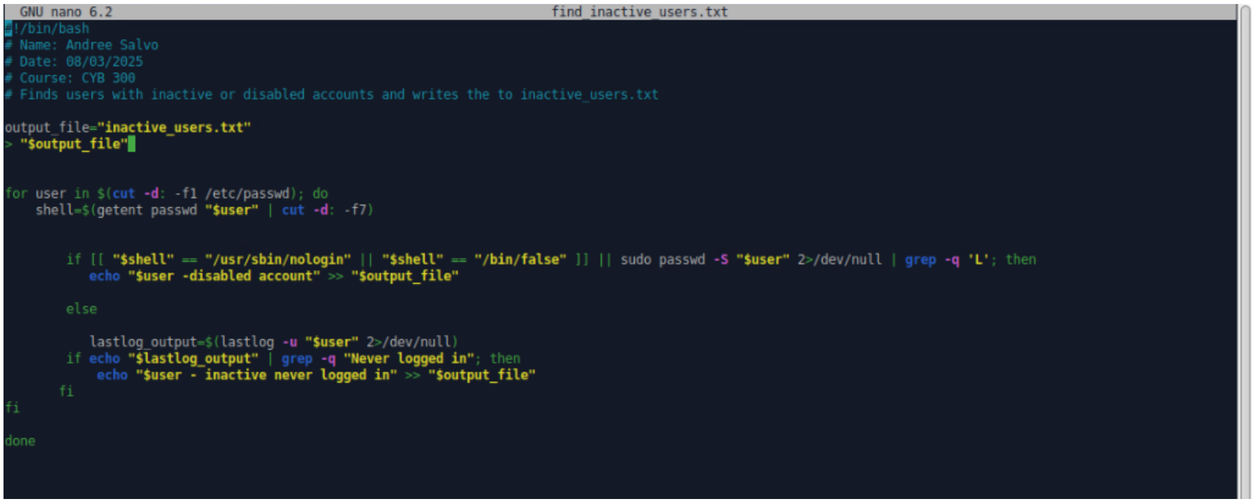
```
#!/bin/bash
# Name: Andree Salvo
# Date: 08/03/2025
# Course: CYB 300
# Adds all HR and Finance users to the Audit group

audit_group="Audit"

# Creating the Audit group if doesn't already exist!

if ! getent group "$audit_group" > /dev/null; then
    sudo groupadd "$audit_group"
    echo "Group $audit_group created."
fi

# Goes through each user on the system!

for user in $(cut -d: -f1 /etc/passwd); do

# Checking if the user is in HR or Finance or not!

if id -nG "$user" | grep -qw "HR" || id -nG "$user" || grep -qw "Finance"; then
    sudo usermod -aG "$audit_group" "$user"
    echo "User $user added to $audit_group."
fi
done
```

Find all users with inactive and disabled accounts across all organizational units and output the list to a text file named inactive_users.txt



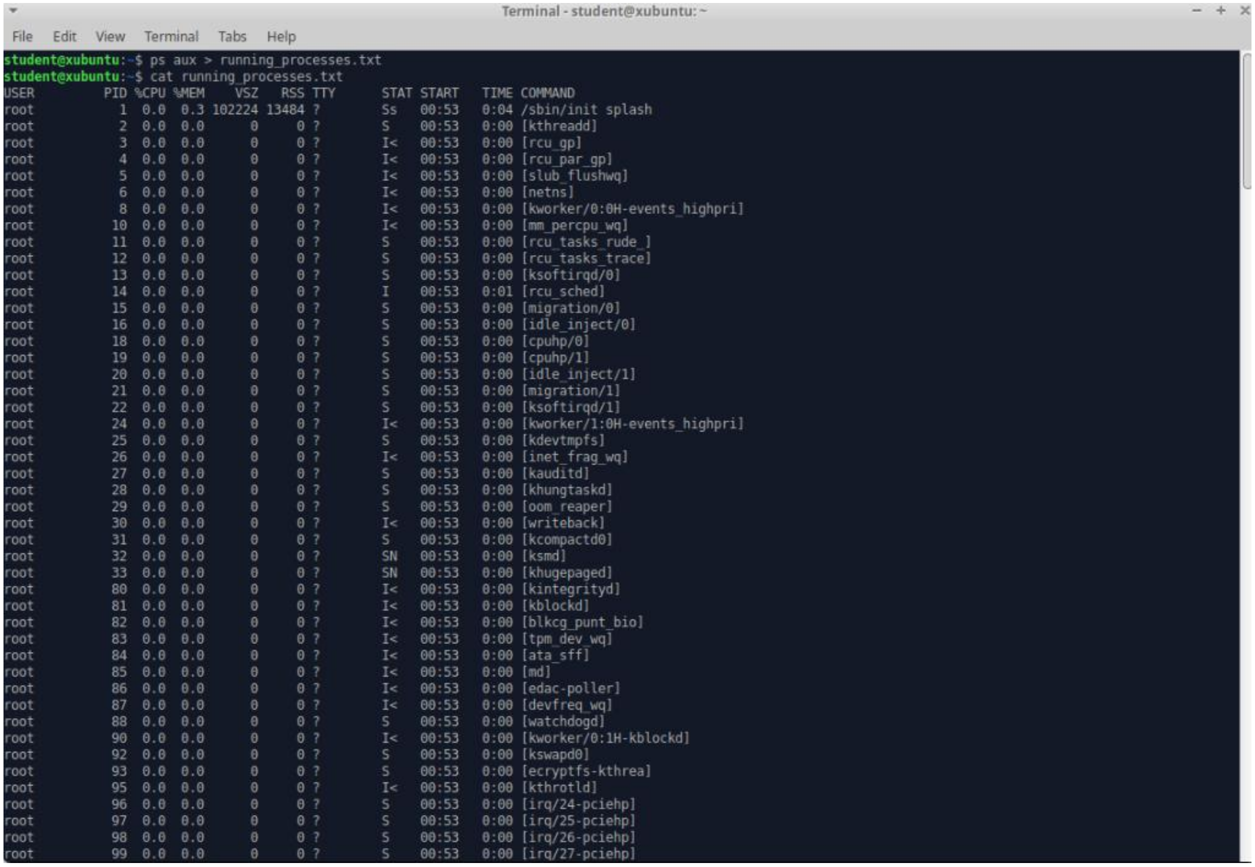
```
student@xubuntu:~$ nano find_inactive_users.sh
student@xubuntu:~$ chmod +x find_inactive_users.sh
student@xubuntu:~$ bash find_inactive_users.sh
[sudo] password for student:
student@xubuntu:~$ cat inactive_users.txt
root - inactive never logged in
daemon -disabled account
bin -disabled account
sys -disabled account
sync -disabled account
games -disabled account
man -disabled account
lp -disabled account
mail -disabled account
news -disabled account
uucp -disabled account
proxy -disabled account
www-data -disabled account
backup -disabled account
list -disabled account
irc -disabled account
gnats -disabled account
nobody -disabled account
systemd-network -disabled account
systemd-resolve -disabled account
messagebus -disabled account
systemd-timesync -disabled account
syslog -disabled account
_apt -disabled account
tss -disabled account
uuidd -disabled account
tcpdump -disabled account
avahi-autoipd -disabled account
usbmux -disabled account
dnsmasq -disabled account
kernoops -disabled account
```

Code:



```
  GNU nano 6.2                                        find_inactive_users.txt
#!/bin/bash
# Name: Andree Salvo
# Date: 08/03/2025
# Course: CYB 300
# Finds users with inactive or disabled accounts and writes the to inactive_users.txt

output_file="inactive_users.txt"
> "$output_file"


for user in $(cut -d: -f1 /etc/passwd); do
    shell=$(getent passwd "$user" | cut -d: -f7)


        if [[ "$shell" == "/usr/sbin/nologin" || "$shell" == "/bin/false" ]] || sudo passwd -S "$user" 2>/dev/null | grep -q 'L'; then
           echo "$user -disabled account" >> "$output_file"

        else

           lastlog_output=$(lastlog -u "$user" 2>/dev/null)
        if echo "$lastlog_output" | grep -q "Never logged in"; then
            echo "$user - inactive never logged in" >> "$output_file"
       fi
fi

done
```

Get a list of all running processes and output the list to a text file named running_processes.txt



```
student@xubuntu:~$ ps aux > running_processes.txt
student@xubuntu:~$ cat running_processes.txt
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.3 102224 13484 ?        Ss   00:53   0:04 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    00:53   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   00:53   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   00:53   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   00:53   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   00:53   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   00:53   0:00 [kworker/0:0H-events_highpri]
root          10  0.0  0.0      0     0 ?        I<   00:53   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        S    00:53   0:00 [rcu_tasks_rude_]
root          12  0.0  0.0      0     0 ?        S    00:53   0:00 [rcu_tasks_trace]
root          13  0.0  0.0      0     0 ?        S    00:53   0:00 [ksoftirqd/0]
root          14  0.0  0.0      0     0 ?        I    00:53   0:01 [rcu_sched]
root          15  0.0  0.0      0     0 ?        S    00:53   0:00 [migration/0]
root          16  0.0  0.0      0     0 ?        S    00:53   0:00 [idle_inject/0]
root          18  0.0  0.0      0     0 ?        S    00:53   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    00:53   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    00:53   0:00 [idle_inject/1]
root          21  0.0  0.0      0     0 ?        S    00:53   0:00 [migration/1]
root          22  0.0  0.0      0     0 ?        S    00:53   0:00 [ksoftirqd/1]
root          24  0.0  0.0      0     0 ?        I<   00:53   0:00 [kworker/1:0H-events_highpri]
root          25  0.0  0.0      0     0 ?        S    00:53   0:00 [kdevtmpfs]
root          26  0.0  0.0      0     0 ?        I<   00:53   0:00 [inet_frag_wq]
root          27  0.0  0.0      0     0 ?        S    00:53   0:00 [kauditd]
root          28  0.0  0.0      0     0 ?        S    00:53   0:00 [khungtaskd]
root          29  0.0  0.0      0     0 ?        S    00:53   0:00 [oom_reaper]
root          30  0.0  0.0      0     0 ?        I<   00:53   0:00 [writeback]
root          31  0.0  0.0      0     0 ?        S    00:53   0:00 [kcompactd0]
root          32  0.0  0.0      0     0 ?        SN   00:53   0:00 [ksmd]
root          33  0.0  0.0      0     0 ?        SN   00:53   0:00 [khugepaged]
root          80  0.0  0.0      0     0 ?        I<   00:53   0:00 [kintegrityd]
root          81  0.0  0.0      0     0 ?        I<   00:53   0:00 [kblockd]
root          82  0.0  0.0      0     0 ?        I<   00:53   0:00 [blkcg_punt_bio]
root          83  0.0  0.0      0     0 ?        I<   00:53   0:00 [tpm_dev_wq]
root          84  0.0  0.0      0     0 ?        I<   00:53   0:00 [ata_sff]
root          85  0.0  0.0      0     0 ?        I<   00:53   0:00 [md]
root          86  0.0  0.0      0     0 ?        I<   00:53   0:00 [edac-poller]
root          87  0.0  0.0      0     0 ?        I<   00:53   0:00 [devfreq_wq]
root          88  0.0  0.0      0     0 ?        S    00:53   0:00 [watchdogd]
root          90  0.0  0.0      0     0 ?        I<   00:53   0:00 [kworker/0:1H-kblockd]
root          92  0.0  0.0      0     0 ?        S    00:53   0:00 [kswapd0]
root          93  0.0  0.0      0     0 ?        S    00:53   0:00 [ecryptfs-kthrea]
root          95  0.0  0.0      0     0 ?        I<   00:53   0:00 [kthrotld]
root          96  0.0  0.0      0     0 ?        S    00:53   0:00 [irq/24-pciehp]
root          97  0.0  0.0      0     0 ?        S    00:53   0:00 [irq/25-pciehp]
root          98  0.0  0.0      0     0 ?        S    00:53   0:00 [irq/26-pciehp]
root          99  0.0  0.0      0     0 ?        S    00:53   0:00 [irq/27-pciehp]
```