SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya
TRUST

K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)
Department of Computer Engineering

Batch:C2 Roll No.: 16010121110

Experiment No. 02

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE: Shell Programming and system calls

AIM: To study the shell script and write the program using shell.

Expected Outcome of Experiment:

CO 1. To introduce basic concepts and functions of operating systems.

Books/ Journals/ Websites referred:

1. **Silberschatz A., Galvin P., Gagne G. "Operating Systems Principles", Willey Eight edition.**
2. **William Stallings "Operating Systems" Person, Seventh Edition Edition.**
3. **Sumitabha Das " UNIX Concepts & Applications", McGraw Hill Second Edition.**

Pre Lab/ Prior Concepts:

The shell provides you with an interface to the UNIX system. It gathers input from you and executes programs based on that input. When a program finishes executing, it displays that program's output.

Shell Scripts

The basic concept of a shell script is a list of commands, which are listed in the order of execution. A good shell script will have comments, preceded by a pound sign, #, describing the steps.

Steps to create a Shell Script:

create a file using any text editor say vi, gedit, nano etc
1.$ vi filename

2.Insert the script/ commands in file and save the file to execute the file we need to give execute permission to the file
3.$ chmod 775 filename
4.Now execute the above file using any of following methods:
$ sh filename
OR
$ ./filename

NOTE: Before adding anything to your script, you need to alert the system that a shell script is being started. This is done using the shebang construct. For example −
#!/bin/sh.

Description of the application to be implemented:

1. Write a shell Script that accepts two file names as command line arguments and compare two file contents and check whether contents are same or not. If they are same, then delete second file

2. Write a shell script that accepts integer and find the factorial of number.

3. Write a shell script for adding users.

4. Write a shell script for counting no of logged in users.

5. Write a shell script for counting no of processes running on system

Program for System Call:

1. Write a Program for creating process using System call (E.g fork())
   Create a child process. Display the details about that process using getpid and getppid functions. In a child process, Open the file using file system calls and read the contents and display.

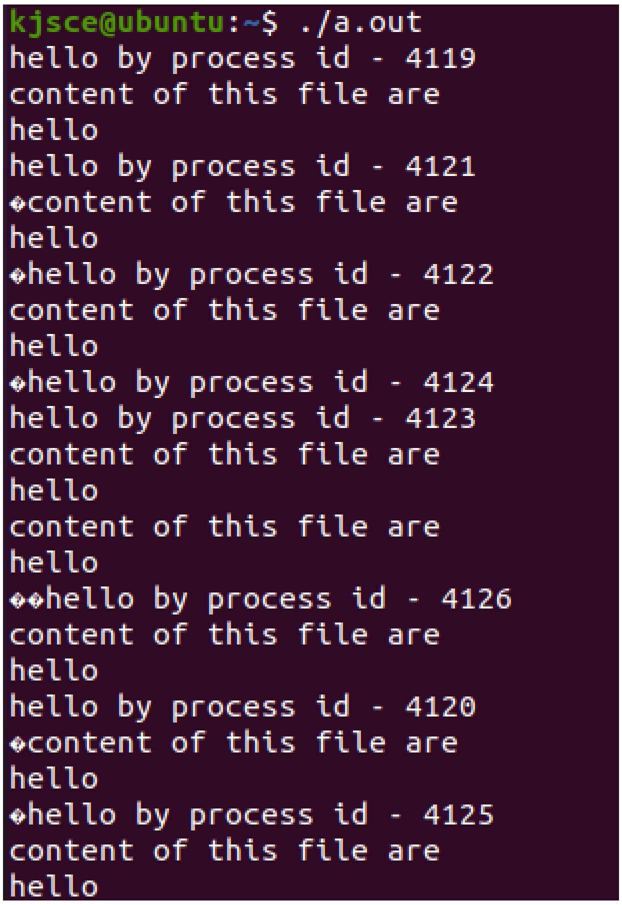
```
Code #include <stdio.h>

#include <sys/types.h>

#include <unistd.h>
```

```
int main()
{        FILE* ptr;
         char ch;


   fork();
   fork();
   fork();
   printf("hello by process id - ");
   printf("%d \n",getpid());
   // Opening file in reading mode
         ptr = fopen("txt1.txt", "r");


         if (NULL == ptr) {
         printf("file can't be opened \n");
         }


         printf("content of this file are \n");


         // Printing what is written in file
         // character by character using loop.
         do {
         ch = fgetc(ptr);
         printf("%c", ch);
```

```
        // Checking if character is not EOF.
        // If it is EOF stop reading.
        } while (ch != EOF);

        // Closing the file
        fclose(ptr);
    return 0;
}
```

Output

here we can see every process opens and closes the file the process id is visible in all there are 2^3=8 processes sometimes the process and output order differs. This is due to C language



```
kjsce@ubuntu:~$ ./a.out
hello by process id - 4119
content of this file are
hello
hello by process id - 4121
�content of this file are
hello
�hello by process id - 4122
content of this file are
hello
�hello by process id - 4124
hello by process id - 4123
content of this file are
hello
content of this file are
hello
��hello by process id - 4126
content of this file are
hello
hello by process id - 4120
�content of this file are
hello
�hello by process id - 4125
content of this file are
hello
```

2.

Implementation details: (printout of code / screen shot)

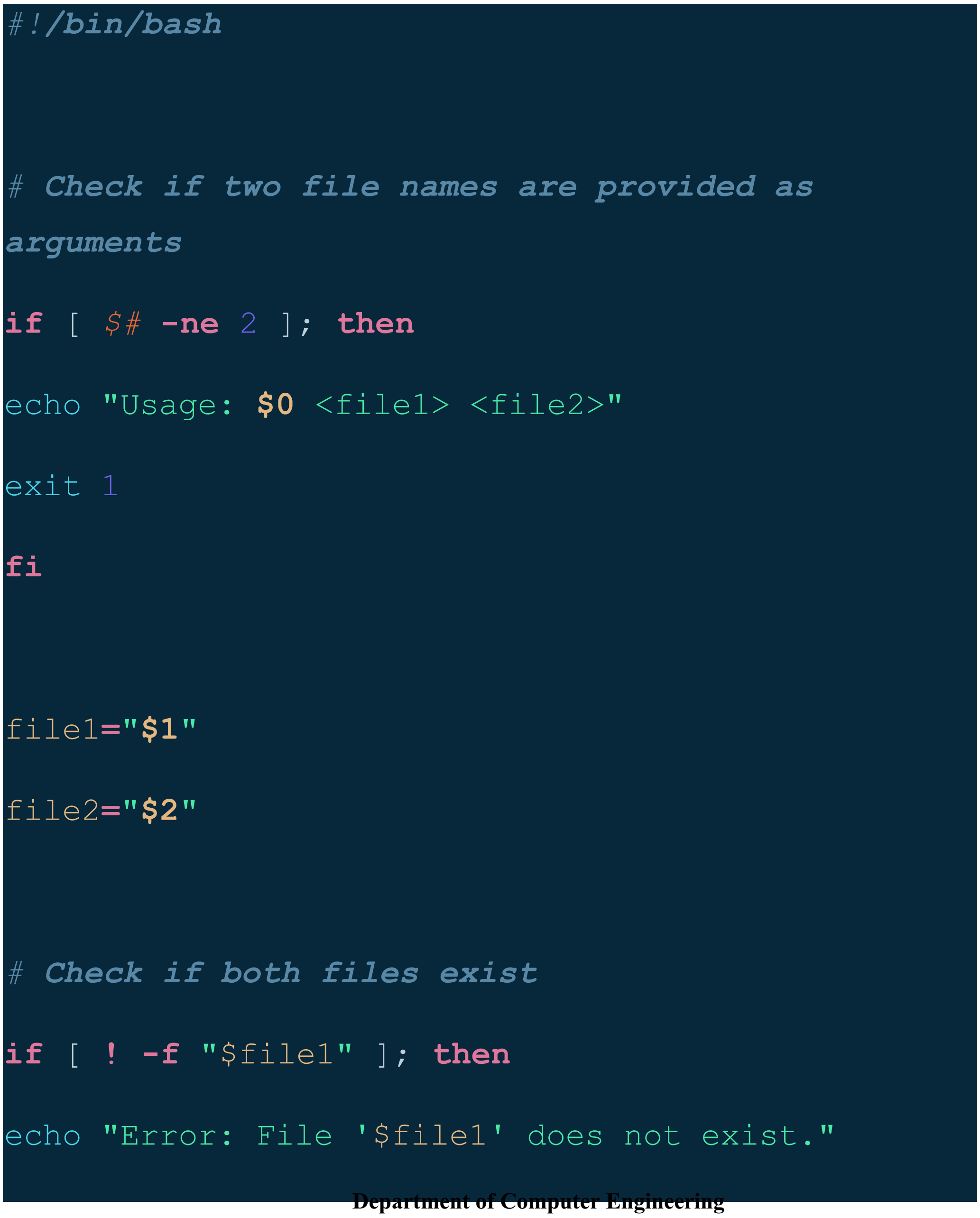
Hello world program

```
#!/bin/bash

echo "hello world"
```

```
kjsce@ubuntu:~$ which bash

/usr/bin/bash

kjsce@ubuntu:~$ cd Desktop

kjsce@ubuntu:~/Desktop$ touch exp2.sh

kjsce@ubuntu:~/Desktop$ ls

exp1  exp2.sh  F2.txt  folder  Smit-2  Smit_OS  Smit_OS_Lab

kjsce@ubuntu:~/Desktop$ ls -al

total 32

drwxr-xr-x  6 kjsce kjsce 4096 Jul 24 16:23 .

drwxr-xr-x 17 kjsce kjsce 4096 May 24 20:04 ..

-rw-rw-r--  1 kjsce kjsce   67 Jul 17 16:52 exp1

-rw-rw-r--  1 kjsce kjsce    0 Jul 24 16:23 exp2.sh

-rw-rw-r--  1 kjsce kjsce   13 Jul 18 15:50 F2.txt

drwxrwxr-x  2 kjsce kjsce 4096 Jul 17 17:36 folder

drwxrwxr-x  2 kjsce kjsce 4096 Jul 18 14:45 Smit-2

drwxrwxr-x  2 kjsce kjsce 4096 Jul 18 15:16 Smit_OS

drwxrwxr-x  2 kjsce kjsce 4096 Jul 18 15:51 Smit_OS_Lab
```

```
kjsce@ubuntu:~/Desktop$ ./exp2.sh

bash: ./exp2.sh: Permission denied

kjsce@ubuntu:~/Desktop$ chmod +x exp2.sh

kjsce@ubuntu:~/Desktop$ ./exp2.sh

hello world
```

Write a shell Script that accepts two file names as command line arguments and compare two file contents and check whether contents are same or not. If they are same, then delete second file



```
#!/bin/bash


# Check if two file names are provided as
arguments

if [ $# -ne 2 ]; then

echo "Usage: $0 <file1> <file2>"

exit 1

fi


file1="$1"

file2="$2"


# Check if both files exist

if [ ! -f "$file1" ]; then

echo "Error: File '$file1' does not exist."
```

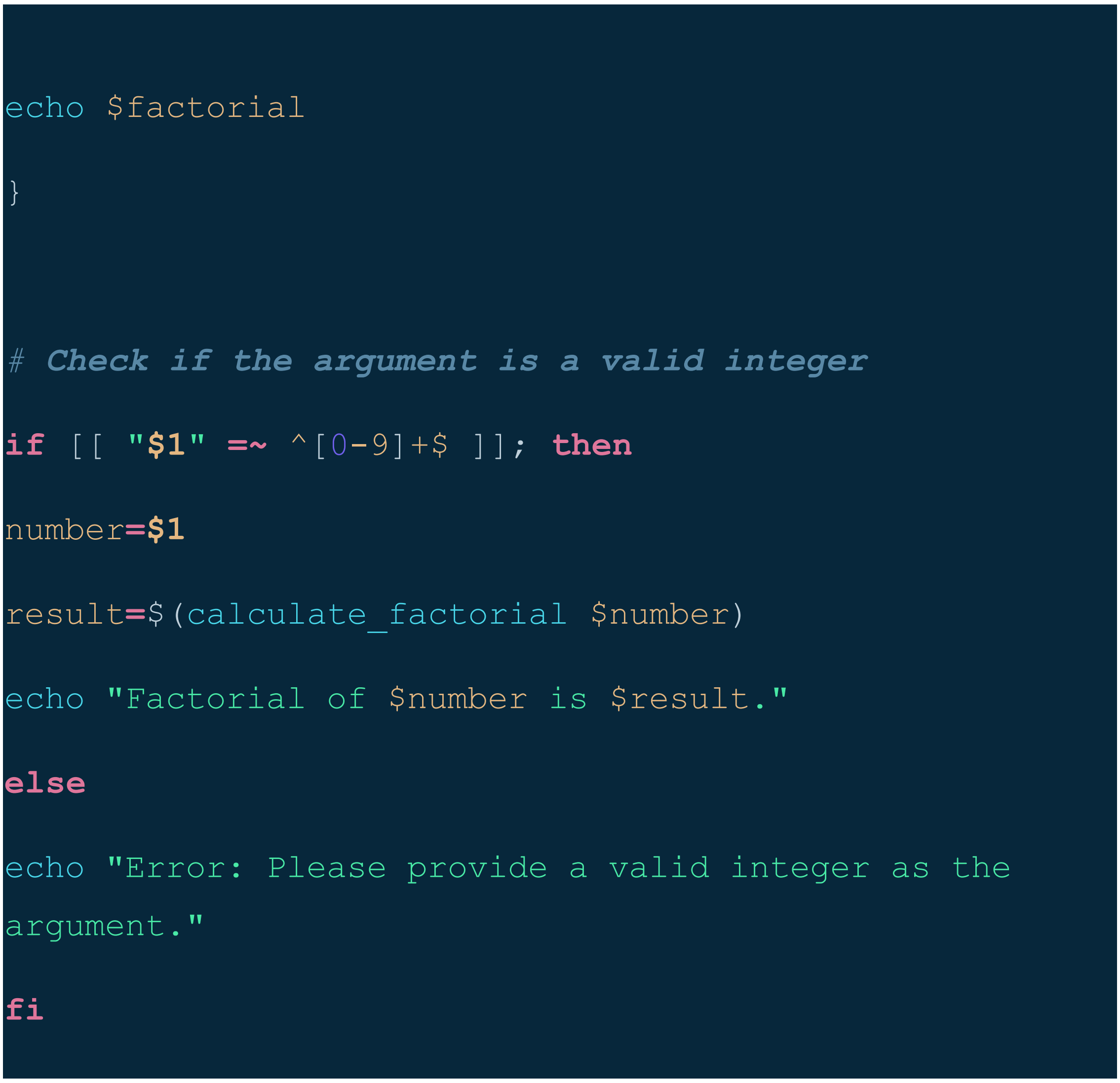
```
exit 1

fi



if [ ! -f "$file2" ]; then

echo "Error: File '$file2' does not exist."

exit 1

fi



# Compare the contents of the files

if cmp -s "$file1" "$file2"; then

echo "The contents of '$file1' and '$file2' are
the same. Deleting '$file2'..."

rm "$file2"

else

echo "The contents of '$file1' and '$file2' are
different."

fi
```

```
kjsce@ubuntu:~/Desktop$ touch a

kjsce@ubuntu:~/Desktop$ touch b

kjsce@ubuntu:~/Desktop$ ./exp2.sh
```

```
Usage: ./exp2.sh <file1> <file2>

kjsce@ubuntu:~/Desktop$ ./exp2.sh a c

Error: File 'c' does not exist.

kjsce@ubuntu:~/Desktop$ ./exp2.sh a b

The contents of 'a' and 'b' are the same. Deleting 'b'...

kjsce@ubuntu:~/Desktop$
```

Write a shell script that accepts integer and find the factorial of number.

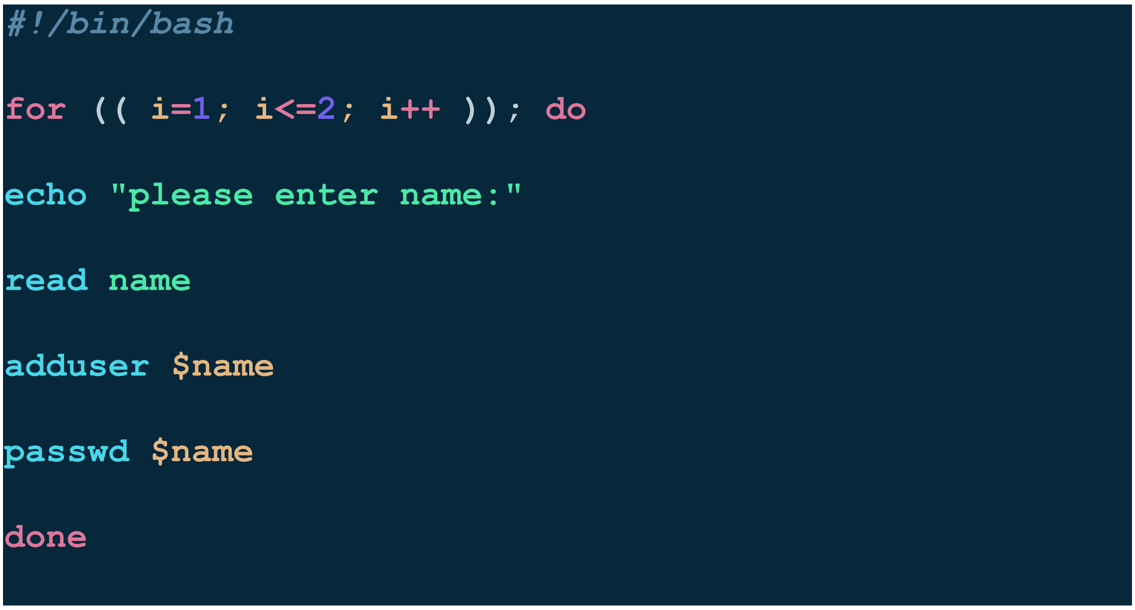
```
#!/bin/bash

# Check if an integer is provided as a command-line argument
if [ $# -ne 1 ]; then
echo "Usage: $0 <integer>"
exit 1
fi

# Function to calculate factorial
calculate_factorial() {
local num=$1
local factorial=1

for (( i=1; i<=num; i++ )); do
factorial=$(( factorial * i ))
done
```



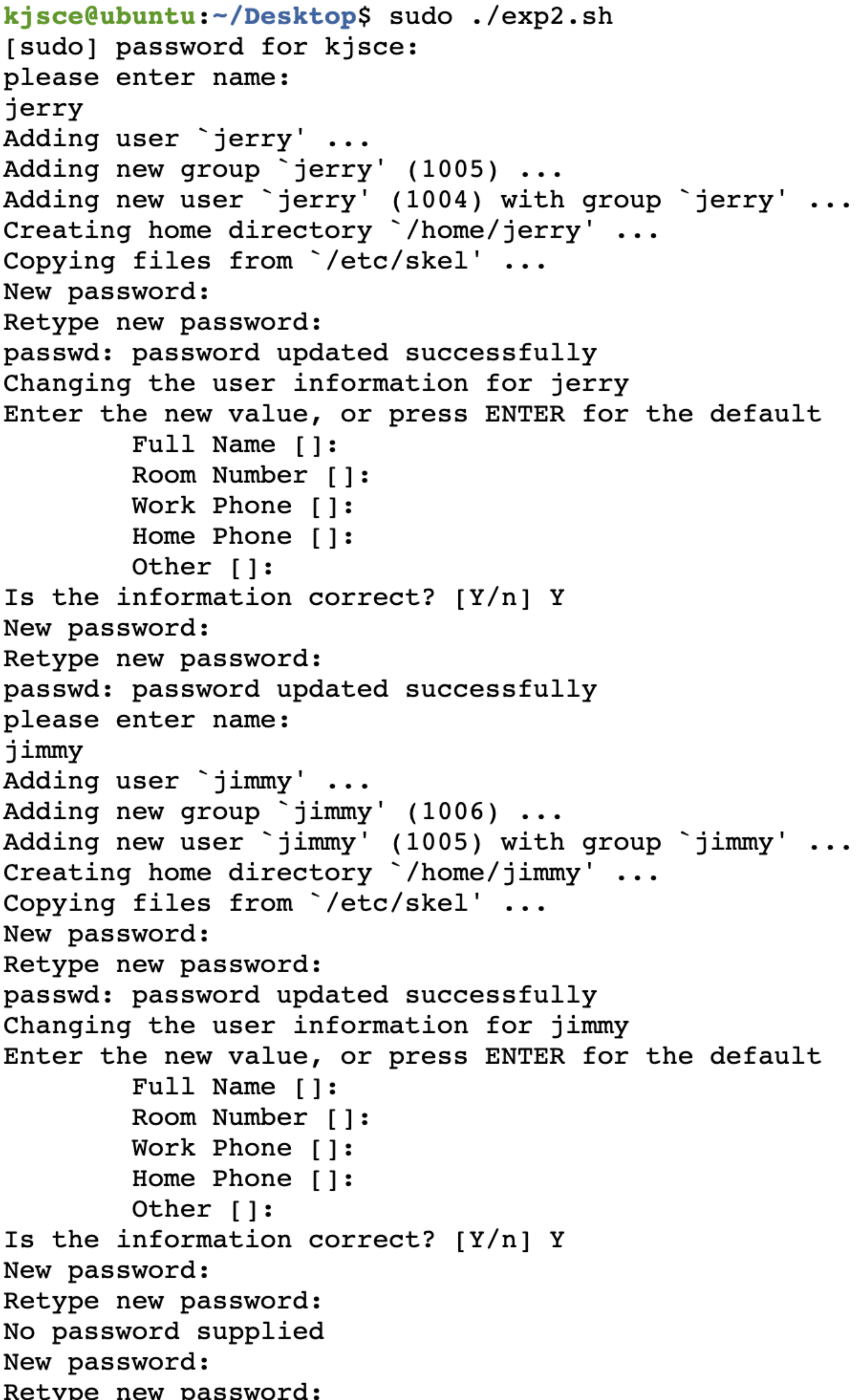
```
echo $factorial

}


# Check if the argument is a valid integer

if [[ "$1" =~ ^[0-9]+$ ]]; then

number=$1

result=$(calculate_factorial $number)

echo "Factorial of $number is $result."

else

echo "Error: Please provide a valid integer as the
argument."

fi
```

```
kjsce@ubuntu:~/Desktop$ ./exp2.sh 5

Factorial of 5 is 120.

kjsce@ubuntu:~/Desktop$ ./exp2.sh 50

Factorial of 50 is -3258495067890909184.
```

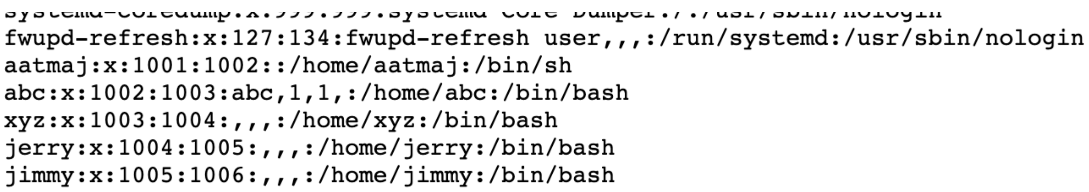
Write a shell script for adding users.



```
#!/bin/bash

for (( i=1; i<=2; i++ )); do

echo "please enter name:"

read name

adduser $name

passwd $name

done
```

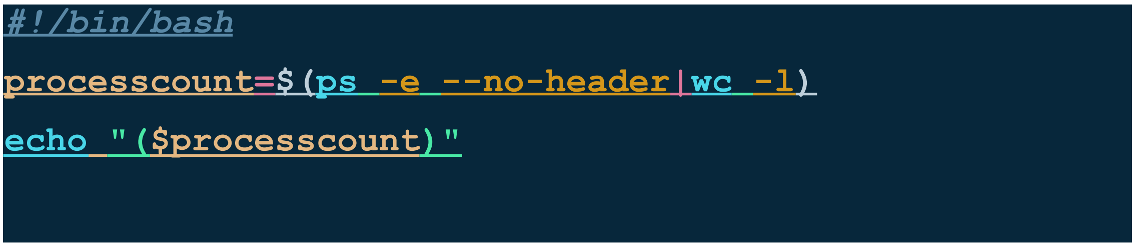


```
kjsce@ubuntu:~/Desktop$ sudo ./exp2.sh
[sudo] password for kjsce:
please enter name:
jerry
Adding user `jerry' ...
Adding new group `jerry' (1005) ...
Adding new user `jerry' (1004) with group `jerry' ...
Creating home directory `/home/jerry' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for jerry
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
New password:
Retype new password:
passwd: password updated successfully
please enter name:
jimmy
Adding user `jimmy' ...
Adding new group `jimmy' (1006) ...
Adding new user `jimmy' (1005) with group `jimmy' ...
Creating home directory `/home/jimmy' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for jimmy
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
New password:
Retype new password:
No password supplied
New password:
Retype new password:
```



```
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
fwupd-refresh:x:127:134:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
aatmaj:x:1001:1002::/home/aatmaj:/bin/sh
abc:x:1002:1003:abc,1,1,:/home/abc:/bin/bash
xyz:x:1003:1004:,,,:/home/xyz:/bin/bash
jerry:x:1004:1005:,,,:/home/jerry:/bin/bash
jimmy:x:1005:1006:,,,:/home/jimmy:/bin/bash
```

Write a shell script for counting no of processes.

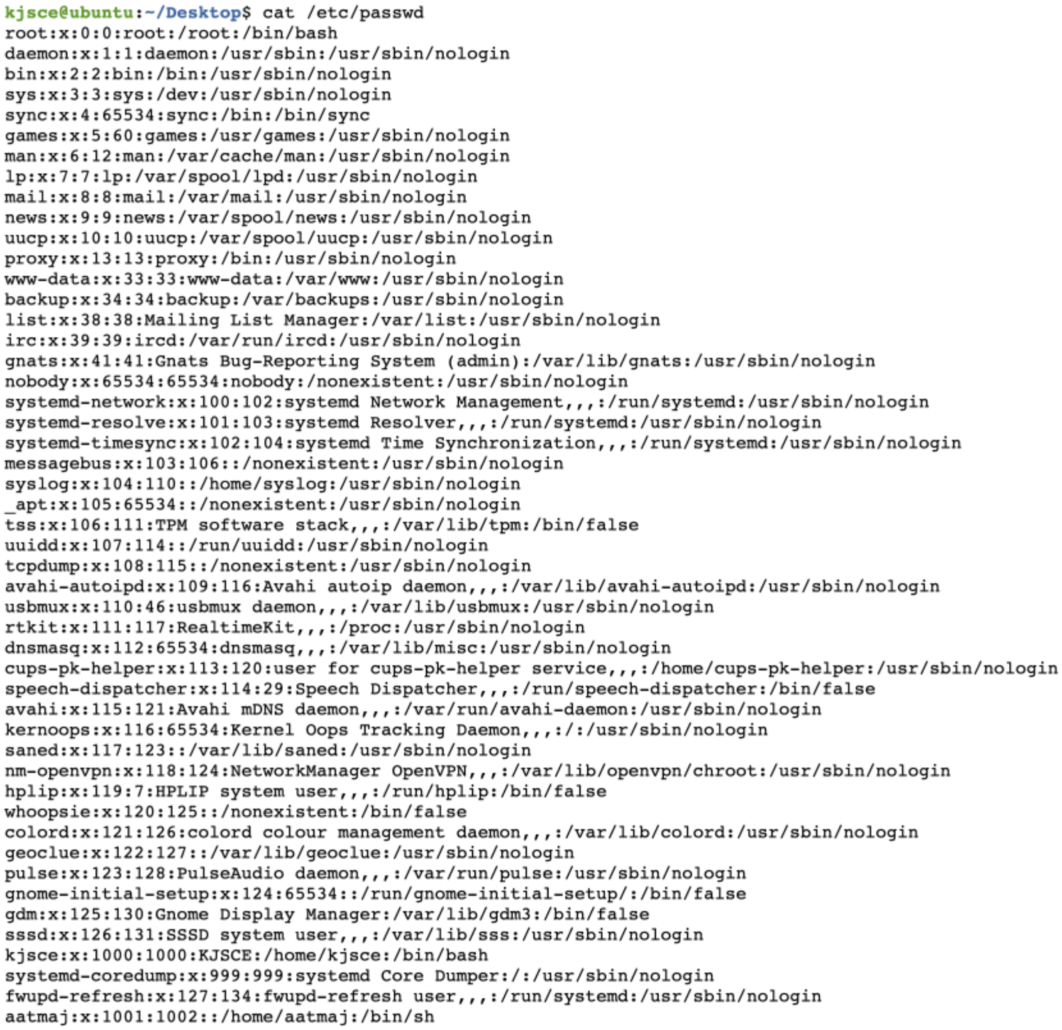


```bash
#!/bin/bash
processcount=$(ps -e --no-header|wc -l)
echo "($processcount)"
```

```
kjsce@ubuntu:~/Desktop$ ./exp2.sh
(232)
```

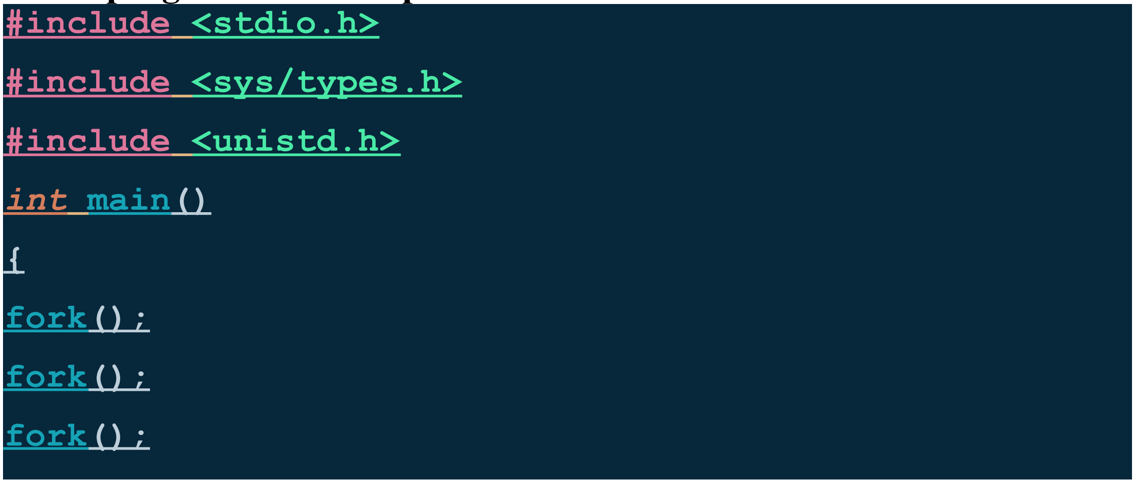
Write a shell script for counting no of Users.

We use cat /etc/passwd to get list of the number of users



```
kjsce@ubuntu:~/Desktop$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/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
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:117:123::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125::/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
sssd:x:126:131:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
kjsce:x:1000:1000:KJSCE:/home/kjsce:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
fwupd-refresh:x:127:134:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
aatmaj:x:1001:1002::/home/aatmaj:/bin/sh
```

Write program to fork a process



```
#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

int main()

{

fork();

fork();

fork();
printf("hello by process id - ");
printf("%d \n",getpid());
return 0;
}
```

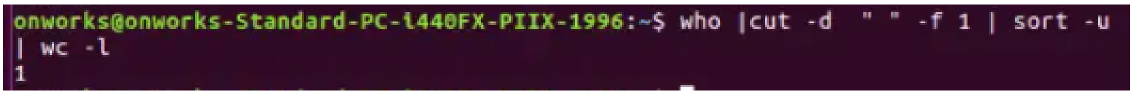
```
kjsce@ubuntu:~/Desktop$ gcc fork.c
kjsce@ubuntu:~/Desktop$ ./a.out
hello by process id - 7634
hello by process id - 7635
hello by process id - 7637
kjsce@ubuntu:~/Desktop$ hello by process id - 7636
hello by process id - 7639
hello by process id - 7640
hello by process id - 7638
hello by process id - 7641
```

Write a program to count the number of users

who | cut -d “ “ -f 1 | sort -u

Logic -

who - shows total number of sessions
cut - We can use the cut command to strip all information except for the user name
-d “ “ is used for seperating using space
sort -u -. This sorts the names alphabetically and filters out any lines that are not unique
wc -l to count the answer

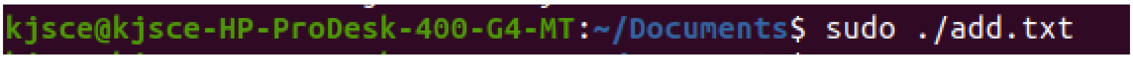


```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ who |cut -d  " " -f 1 | sort -u
| wc -l
1
```

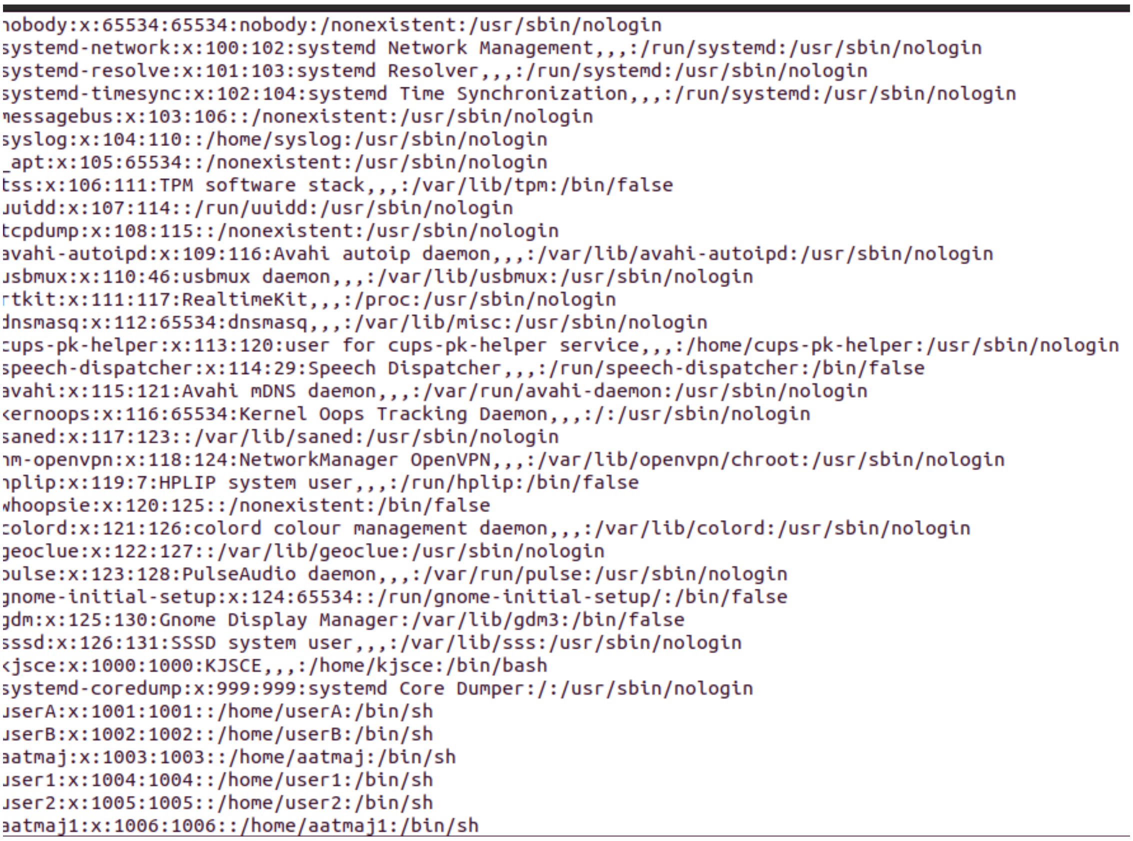
Write a program to add users using input redirection

Logic - read file linmes one by one, get usernames and then add users

```
# Input file containing usernames and passwords
input_file="user_data.txt"
while IFS= read -r line; do
  username=$(echo "$line" | cut -d " " -f 1)
  password=$(echo "$line" | cut -d " " -f 2)
  useradd -m "$username" -p "$password"
  done < "$input_file"
```



```
kjsce@kjsce-HP-ProDesk-400-G4-MT:~/Documents$ sudo ./add.txt
```



```
nobody:x:65534:65534:nobody:/nonexistent:/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
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:117:123::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125::/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
sssd:x:126:131:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
kjsce:x:1000:1000:KJSCE,,,:/home/kjsce:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
userA:x:1001:1001::/home/userA:/bin/sh
userB:x:1002:1002::/home/userB:/bin/sh
aatmaj:x:1003:1003::/home/aatmaj:/bin/sh
user1:x:1004:1004::/home/user1:/bin/sh
user2:x:1005:1005::/home/user2:/bin/sh
aatmaj1:x:1006:1006::/home/aatmaj1:/bin/sh
```

In output, new usernames are visible when we use ./etc/passwds file

Conclusion:

Thus we have understood how shell code works in Linux systems. Shell commands are commands that can be executed from the terminal or a file. They do not need any programming language as they are built in within Linux. We learnt the shell language basics. This will enable us to do various tks like interact with servers and write viruses.

We understood how code works for system-specific tasks like finding out the number of processes or general tasks like finding the factorial of a number.

Post Lab Descriptive Questions

1. What are the different types of commonly used shells on a typical linux system?

- Bourne Shell (sh) The Bourne shell was the first default shell on Unix systems, released in 1979.
- C Shell (csh)
- TENEX C Shell (tcsh)
- KornShell (ksh)
- Debian Almquist Shell (dash)
- Bourne Again Shell (bash)
- Z Shell (zsh)
- Friendly Interactive Shell (fish)

2. How do you find out what's your shell?

```
printf "My current shell - %s\n" "$SHELL"
```

3. List the advantages and disadvantages of shell scripting.

Advantages of shell scripting:

- Automation of repetitive tasks.
- Quick and easy prototyping of commands and processes.
- Seamless integration with system utilities and pipelines.

Disadvantages of shell scripting:

- Limited performance for complex computations.
- Lack of strong data structures and object-oriented features.
- Platform-dependent syntax variations across different shells.

Date: 5 Aug 2023 **Signature of faculty in-charge**