# Shri G.S Institute of Technology & Science
# Operating Systems
# Assignment 3 – INDEX

| S,NO | ASSIGNMENT-3 | CO | BL |
|---|---|---|---|
| 1. | **Linux shell script program to swap two numbersSwapping using third variable and without third VARIABLE.** | 1 | 2,3 |
| 2. | **Linux shell script program to demonstrate the '$#' variable and Linux shell script program to print thecurrent process id** | 1 | 3,4 |
| 3. | **Linux shell script to demonstrate comparision operators** | 1 | 4 |
| 4. | **Write a Shell Script Program to Checking if a File Exists or Not.** | 1 | 4 |
| 5. | **Write a Shell script to determine if a number is positive, negative, or zero** | 1 | 1,2 |
| 6. | **Write a shell script to manage user accounts on a Linux system. The script should allow the administrator to create, delete, or modify user accounts based on user input.** | 1 | 2,3 |
| 7. | **Create a Sample Shell Script that removes in the Current Directory that are older than 4 days based on their file name .** | 1 | 3,4 |
| 8. | **Create a script that counts the number of words, lines, and characters in a 2 text files.** | 1 | 4,5 |
| 9. | **Write a Shell Script programs using Switch Case and read the following instruction for that Programs :**<br>1. Create a custom menu using echo statement and show the menu<br>2. Create an infinite loop using while statement that accept the user input option and generate the output continuously until the user input matches the exit pattern.<br>3. Take input from the user using read statement and store it in a variable.<br>4. Use case statement to check if the input matches with the pattern.<br>5. Create custom pattern.<br>6. Exit the case statement using esac keyword.<br>And Output Show Like that : | 1 | 3,2 |
| | SELECT YOUR FAVORITE FRUIT<br>1. Apple<br>2. Grapes<br>3. Mango<br>4. Exit from menu<br>Enter your menu choice [1-4]: 1<br>You have selected the option 1<br>Selected Fruit is Apple.<br>Enter your menu choice [1-4]: 2<br>You have selected the option 2<br>Selected Fruit is Grapes.<br>Enter your menu choice [1-4]: 3<br>You have selected the option 3<br>Selected Fruit is Mango.<br>Enter your menu choice [1-4]: 4<br>Quiting ... | | |
| 10. | **Write a Shell Script Programs to take input a Char from User and check the Char is Lower case ,Upper Case and Vowels & Consonants .** | 1 | 3,2 |
| 11. | **Write a shell script to find out the greatest among three inputs.** | 1 | 3,2 |

| | | | |
|---|---|---|---|
| 12. | **Write a shell script to calculate the net salary of an employee in a particular month considering various allowances (TA, DA, HRA) and deductions (INCOME TAX, PROVIDEND FUND) as:**<br>**a. TA=15 percent of basic salary**<br>**b. DA=2 percent of basic salary**<br>**c. HRA=10 percent of basic salary**<br>**d. INCOME TAX=5 percent of salary**<br>**e. PROVIDEND FUND=10 percent of salary** | 1 | 3,2 |
| 13. | **A departmental store announces its festival scheme to customers on cash payment. The scheme is asfollows-**<br>**a. If purchase amount is less than 1000 then Tax=2% and discount=10%.**<br>**b. If purchase amount is greater than 1000 then Tax=5 % and discount=20%.** | 1 | 3,2 |
| 14. | **Write a C/JAVA Program for system calls of Unix operating systems (opendir, readdir, closedir)** | 1 | 3,2 |
| 15. | **Write a C/JAVA Program for Process system calls of Unix operating systems (fork, getpid, exit)** | 1 | 3,2 |
| 16. | **Write a C/JAVA Program to implement the system calls wait ( ) and exit ( )** | 1 | 3,2 |
| 17. | **Write a C/JAVA Program to implement the system call execl ( ).** | 1 | 3,2 |
| 18. | **Write a C/JAVA Program to implement the system call execv ( )** | 1 | 3,2 |
| 19. | **write a 'c' program for I/O system calls.** | 1 | 3,2 |

1.) Linux shell script program to swap two numbers Swapping using third variable and without third VARIABLE.

```
#!/bin/bash
echo "Enter first number: "
read n1
echo "Enter second number: "
read n2
echo "Before swapping: n1 = $n1, n2 = $n2"
n1=$((n1 ^ n2))
n2=$((n1 ^ n2))
n1=$((n1 ^ n2))
echo "After swapping: n1 = $n1, n2 = $n2"
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
➜ ./swap.sh
Enter first number:
12
Enter second number:
10
Before swapping: n1 = 12, n2 = 10
After swapping: n1 = 10, n2 = 12
```

2.) Linux shell script program to demonstrate the '$#' variable and Linux shell script program to print thecurrent process id

```
#!/bin/bash
echo "Number of arguments passed: $#"
echo "Current Process ID: $$"
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 took 4.0s …
➜ ./process_id.sh 10 38 43
Number of arguments passed: 3
Current Process ID: 8257
```

3.) Linux shell script to demonstrate comparision operators

```
#!/bin/bash
echo "Enter first number: "
read num1
echo "Enter second number: "
read num2
if [ $num1 -eq $num2 ]; then
   echo "The numbers are equal."
else
   echo "The numbers are not equal."
fi
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
➜ ./cm_operator.sh
Enter first number:
90
Enter second number:
34
The numbers are not equal.
```

4.) Write a Shell Script Program to Checking if a File Exists or Not

```
#!/bin/bash
echo "Enter the file name: "
read filename
if [ -e "$filename" ]; then
   echo "The file '$filename' exists."
else
   echo "The file '$filename' does not exist."
fi
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
➜ ./file_exists.sh
Enter the file name:
cm_operator.sh
The file 'cm_operator.sh' exists.
```

5.) Write a Shell script to determine if a number is positive, negative, or zero

```
#!/bin/bash
echo "Enter a number: "
read num
if [ $num -gt 0 ]; then
   echo "The number $num is positive."
elif [ $num -lt 0 ]; then
   echo "The number $num is negative."
else
   echo "The number is zero."
fi
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 took 18.7s …
➜ ./num_type.sh
Enter a number:
-30
The number -30 is negative.
```

6.) Write a shell script to manage user accounts on a Linux system. The script should allow the administrator to create, delete, or modify user accounts based on user input.

```
#!/bin/bash
create_user() {
   read -p "Enter username to create: " username
   if id "$username" &>/dev/null; then
      echo "User '$username' already exists!"
```

```
    else
        sudo useradd "$username"
        echo "User '$username' created successfully!"
        sudo passwd "$username"
    fi
}

delete_user() {
    read -p "Enter username to delete: " username
    if id "$username" &>/dev/null; then
        sudo userdel -r "$username"
        echo "User '$username' deleted successfully!"
    else
        echo "User '$username' does not exist!"
    fi
}

modify_user() {
    read -p "Enter username to modify: " username
    if id "$username" &>/dev/null; then
        echo "1. Change Password"
        echo "2. Lock User"
        echo "3. Unlock User"
        echo "4. Add User to Group"
        read -p "Choose an option: " choice

        case $choice in
            1) sudo passwd "$username" ;;
            2) sudo usermod -L "$username"; echo "User '$username' locked!" ;;
            3) sudo usermod -U "$username"; echo "User '$username' unlocked!" ;;
            4) read -p "Enter group name: " group
               sudo usermod -aG "$group" "$username"
               echo "User '$username' added to group '$group'!" ;;
            *) echo "Invalid option!" ;;
        esac
    else
        echo "User '$username' does not exist!"
    fi
}

while true; do
    echo "User Management Script"
    echo "1. Create User"
    echo "2. Delete User"
    echo "3. Modify User"
    echo "4. Exit"
    read -p "Choose an option: " option

    case $option in
        1) create_user ;;
        2) delete_user ;;
```

```
        3) modify_user ;;
        4) echo "Exiting..."; exit 0 ;;
        *) echo "Invalid option! Try again." ;;
      esac
    done
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 took 23.8s …
⚡ ➜ ./users_manupliation.sh
User Management Script
1. Create User
2. Delete User
3. Modify User
4. Exit
Choose an option: 2
Enter username to delete: newuser
uid=1001(newuser) gid=1001(newuser) groups=1001(newuser)
userdel: newuser mail spool (/var/mail/newuser) not found
User 'newuser' deleted successfully!
```

7.) Create a Sample Shell Script that removes in the Current Directory that are older than 4 days based on their file name.

```
#!/bin/bash
days=4
find . -type f -mtime +$days -exec rm -f {} \;
echo "All files older than $days days have been deleted from the current directory."
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 took 3.6s …
⚡ ➜ ./four_days.sh
All files older than 4 days have been deleted from the current directory.
```

8.) Create a script that counts the number of words, lines, and characters in 2 text files.

```
#!/bin/bash
if [ $# -ne 2 ]; then
   echo "Usage: $0 <file1> <file2>"
   exit 1
fi
file1=$1
file2=$2
if [ ! -f "$file1" ]; then
   echo "Error: File '$file1' does not exist!"
   exit 1
fi
if [ ! -f "$file2" ]; then
   echo "Error: File '$file2' does not exist!"
   exit 1
fi
echo "  w    l    c"
echo "Total for both files:"
wc "$file1" "$file2"
exit 0
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
➜ ./count_two_files.sh file_1.txt file_2.txt
  w    l    c
Total for both files:
  98  658 3125 file_1.txt
  60  387 1993 file_2.txt
 158 1045 5118 total
```

9.) Write a Shell Script programs using Switch Case and read the following instruction for that Programs :

1. Create a custom menu using echo statement and show the menu
2. Create an infinite loop using while statement that accept the user input option and generate the output continuously until the user input matches the exit pattern.
3. Take input from the user using read statement and store it in a variable.
4. Use case statement to check if the input matches with the pattern.
5. Create custom pattern.
6. Exit the case statement using esac keyword.

```
#!/bin/bash
echo "SELECT YOUR FAVORITE FRUIT"
echo "1. Apple"
echo "2. Grapes"
echo "3. Mango"
echo "4. Exit from menu"
while true; do
   read -p "Enter your menu choice [1-4]: " choice

   case $choice in
      1)echo "Selected Fruit is Apple."
        ;;
      2) echo "Selected Fruit is Grapes."
        ;;
      3) echo "Selected Fruit is Mango."
        ;;
      4) echo "Quitting ..."
        exit 0
        ;;
      *) echo "Invalid choice! Please enter a number between 1 and 4."
        ;;
   esac
   echo
done
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
⚡ ➜ ./fav_fruit.sh
SELECT YOUR FAVORITE FRUIT
1. Apple
2. Grapes
3. Mango
4. Exit from menu
Enter your menu choice [1-4]: 1
Selected Fruit is Apple.

Enter your menu choice [1-4]: 3
Selected Fruit is Mango.

Enter your menu choice [1-4]: 4
Quitting ...
```

10.) Write a Shell Script Programs to take input a Char from User and check the Char is Lower case ,Upper Case and Vowels & Consonants

```
#!/bin/bash
read -p "Enter a single character: " char
if [[ ${#char} -ne 1 ]]; then
    echo "Error: Please enter only a single character!"
    exit 1
fi
if [[ "$char" =~ [a-z] ]]; then
    echo "Lowercase letter."
    if [[ "$char" =~ [aeiou] ]]; then
        echo "It is a vowel."
    else
        echo "It is a consonant."
    fi
elif [[ "$char" =~ [A-Z] ]]; then
    echo "Uppercase letter."
    if [[ "$char" =~ [AEIOU] ]]; then
        echo "It is a vowel."
    else
        echo "It is a consonant."
    fi
else
    echo "The character '$char' is not a letter."
```

```
fi
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 took 16.5s …
⚡ → ./char_type.sh
Enter a single character: a
Lowercase letter.
It is a vowel.
```

10.)Write a shell script to find out the greatest among three inputs.

```
#!/bin/bash
read -p "Enter first number: " num1
read -p "Enter second number: " num2
read -p "Enter third number: " num3

if (( num1 >= num2 && num1 >= num3 )); then
   echo "The greatest number is: $num1"
elif (( num2 >= num1 && num2 >= num3 )); then
   echo "The greatest number is: $num2"
else
   echo "The greatest number is: $num3"
fi
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 took 9.1s …
⚡ → ./greatest.sh
Enter first number: 12
Enter second number: 45
Enter third number: 89
The greatest number is: 89
```

12.) Write a shell script to calculate the net salary of an employee in a particular month considering various allowances (TA, DA, HRA) and deductions (INCOME TAX, PROVIDEND FUND) as:

a.TA=15 percent of basic salary
b.DA=2 percent of basic salary
c.HRA=10 percent of basic salary
d.INCOME TAX=5 percent of salary
e.PROVIDEND FUND=10 percent of salary

```
#!/bin/bash

read -p "Enter Basic Salary: " basic_salary

TA=$(echo "$basic_salary * 0.15" | bc)

DA=$(echo "$basic_salary * 0.02" | bc)

HRA=$(echo "$basic_salary * 0.10" | bc)

gross_salary=$(echo "$basic_salary + $TA + $DA + $HRA" | bc)
```

```
income_tax=$(echo "$gross_salary * 0.05" | bc)
provident_fund=$(echo "$gross_salary * 0.10" | bc)
net_salary=$(echo "$gross_salary - $income_tax - $provident_fund" | bc)

echo "Basic Salary: $basic_salary"
echo "TA (15%): $TA"
echo "DA (2%): $DA"
echo "HRA (10%): $HRA"
echo "Gross Salary: $gross_salary"
echo "Income Tax (5%): $income_tax"
echo "Provident Fund (10%): $provident_fund"
echo "Net Salary: $net_salary"
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
➜ ./net_salary.sh
Enter Basic Salary: 136000
Basic Salary: 136000
TA (15%): 20400.00
DA (2%): 2720.00
HRA (10%): 13600.00
Gross Salary: 172720.00
Income Tax (5%): 8636.00
Provident Fund (10%): 17272.00
Net Salary: 146812.00
```

13.) A departmental store announces its festival scheme to customers on cash payment. The scheme is asfollows

a.If purchase amount is less than 1000 then Tax=2% and discount=10%.

b.If purchase amount is greater than 1000 then Tax=5 % and discount=20%.

```
#!/bin/bash
read -p "Enter purchase amount: " amount
if (( amount < 1000 )); then
    tax=$(( amount * 2 / 100 ))
    discount=$(( amount * 10 / 100 ))
else
```

```
    tax=$(( amount * 5 / 100 ))
    discount=$(( amount * 20 / 100 ))
fi
final_amount=$(( amount + tax - discount ))
echo "Purchase Amount: $amount"
echo "Tax Applied: $tax"
echo "Discount Applied: $discount"
echo "Final Amount to Pay: $final_amount"
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
➜ ./d_store.sh
Enter purchase amount: 299
Purchase Amount: 299
Tax Applied: 5
Discount Applied: 29
Final Amount to Pay: 275
```

14.) Write a C/JAVA Program for system calls of Unix operating systems (opendir, readdir, closedir)

```
#include <stdio.h>
#include <dirent.h>

int main() {
    struct dirent *entry;
    DIR *dir = opendir(".");

    if (dir == NULL) {
        perror("Unable to open directory");
        return 1;
    }
    printf("Files in the current directory:\n");
    while ((entry = readdir(dir)) != NULL) {
        printf("%s\n", entry->d_name);
    }
    closedir(dir);
```

```
        return 0;
    }
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
⚡ → ./list_files
Files in the current directory:
char_type.sh
four_days.sh
count_two_files.sh
=
list_files.c
swap.sh
fav_fruit.sh
..
cm_operator.sh
d_store.sh
greatest.sh
file_exists.sh
.
users_manupliation.sh
file_2.txt
process_id.sh
list_files
file_1.txt
num_type.sh
net_salary.sh
```

15.) Write a C/JAVA Program for Process system calls of Unix operating systems (fork, getpid, exit)

```
import java.io.File;
public class list_files {
    public static void main(String[] args) {
        File directory = new File("."); // Current directory

        if (!directory.isDirectory()) {
            System.out.println("Error: Not a valid directory.");
            return;
        }

        String[] files = directory.list();
        if (files != null) {
            System.out.println("Files in the current directory:");
            for (String file : files) {
```

```
            System.out.println(file);
        }
    } else {
        System.out.println("Error reading directory.");
    }}}
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
⚡ ➜ java list_files
Files in the current directory:
list_files.java
char_type.sh
four_days.sh
count_two_files.sh
=
list_files.class
swap.sh
fav_fruit.sh
cm_operator.sh
d_store.sh
greatest.sh
file_exists.sh
users_manupliation.sh
file_2.txt
process_id.sh
file_1.txt
num_type.sh
net_salary.sh
```

16.) Write a C/JAVA Program to implement the system calls wait ( ) and exit ( )

```
class Task extends Thread {
    public void run() {
        System.out.println("Task started...");
        try { Thread.sleep(2000); } catch (InterruptedException e) {}
        System.out.println("Task completed.");
        synchronized (this) { notify(); }
    }}
public class wait_exit {
    public static void main(String[] args) throws InterruptedException {
        Task task = new Task();
        task.start();
        synchronized (task) {
            System.out.println("Main thread waiting...");
```

```
            task.wait();
        }
        System.out.println("Exiting program...");
        System.exit(0);
    }}
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
⚡ → java wait_exit
Main thread waiting...
Task started...
Task completed.
Exiting program...
```

17.) Write a C/JAVA Program to implement the system call execl ( ).

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class execc {
    public static void main(String[] args) {
        try {
            System.out.println("Executing ls command...");
            Process process = Runtime.getRuntime().exec("ls");
            BufferedReader reader = new BufferedReader(new
InputStreamReader(process.getInputStream()));
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
            process.waitFor();
            System.out.println("Command executed successfully.");
        } catch (IOException | InterruptedException e) {
            e.printStackTrace();
```

```
        } }
}OUTPUT:
```

```
with root in ~/operating_system/assignment_3 …
⚡ → java execc
Executing ls command...
=
Task.class
all_driectory.class
all_driectory.java
char_type.sh
cm_operator.sh
count_two_files.sh
d_store.sh
execc.class
execc.java
fav_fruit.sh
file_1.txt
file_2.txt
file_exists.sh
four_days.sh
greatest.sh
list_files.class
list_files.java
net_salary.sh
num_type.sh
process_id.sh
swap.sh
users_manupliation.sh
wait_exit.class
wait_exit.java
Command executed successfully.
```

18.) Write a C/JAVA Program to implement the system call execv ( )

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class list_1 {
    public static void main(String[] args) {
        try {
            String[] command = {"/bin/ls", "-l", "/root/operating_system"};
            Process process = Runtime.getRuntime().exec(command);
            BufferedReader reader = new BufferedReader(new
InputStreamReader(process.getInputStream()));
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
```

```
            process.waitFor();
            System.out.println("Command executed successfully.");
        } catch (IOException | InterruptedException e) {
            e.printStackTrace();
        }}}
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
⚡ → java list_l
total 20
drwxr-xr-x 2 root root 4096 Feb 17 22:25 assignment_1
drwxr-xr-x 2 root root 4096 Feb 24 09:48 assignment_2
drwxr-xr-x 2 root root 4096 Mar 23 14:47 assignment_3
drwxr-xr-x 2 root root 4096 Mar  6 18:55 satvik
drwxr-xr-x 2 root root 4096 Mar  6 18:55 shiv
Command executed successfully.
```

19.) Write a ‘c’ program for I/O system calls.

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
void main() {
    int fd;
    char buffer[100];
    // Creating and writing to a file
    fd = open("write.txt", O_CREAT | O_WRONLY, 0644);
    if (fd < 0) {
        perror("Error opening file");
        return;
    }
    write(fd, "Hello, this is a test for I/O system calls.\n", 44);
    close(fd);
    printf("Data written to write.txt\n");
    // Reading from a file
    fd = open("write.txt", O_RDONLY);
    if (fd < 0) {
        perror("Error opening file for reading");
```

```
        return;
    }
    printf("Reading from write.txt:\n");
    int bytes_read = read(fd, buffer, sizeof(buffer) - 1);
    if (bytes_read > 0) {
        buffer[bytes_read] = '\0';
        printf("%s\n", buffer);
    }
    close(fd);
    // Taking user input
    printf("Enter some text: ");
    bytes_read = read(STDIN_FILENO, buffer, sizeof(buffer) - 1);
    buffer[bytes_read - 1] = '\0';  // Remove newline
    //Writing user input to a new file
    fd = open("user_input.txt", O_CREAT | O_WRONLY, 0644);
    if (fd < 0) {
        perror("Error opening file for writing");
        return;
    }
    write(fd, buffer, bytes_read);
    close(fd);
    printf("User input saved to user_input.txt\n");
}
```

OUTPUT:

```
with root in ~/operating_system/assignment_3 …
⚡ → ./system_calls
Data written to write.txt
Reading from write.txt:
Hello, this is a test for I/O system calls.

hello this is shiv
Enter some text: User input saved to user_input.txt
```