

Operating Systems LAB

PAPER CODE : CIC-353

Faculty Name : Ms. Kavita Saxena

Name : Amit Singhal

Enrollment No. : 11614802722

Branch : Computer Science & Engg.

Semester | Group : 5C6



MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY
PSP Area, Plot No. 1, Sector-22, Rohini, Delhi-110086



MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

Computer Science & Engineering Department

VISION

"To be centre of excellence in education, research and technology transfer in the field of computer engineering and promote entrepreneurship and ethical values."

MISSION

To foster an open, multidisciplinary and highly collaborative research environment for producing world-class engineers capable of providing innovative solutions to real-life problems and fulfil societal needs.

LAB Assessment Sheet

[illegible]

[illegible]

Lab Exercise - 1

❖ AIM :: Introduction to Linux & vi-Editor

1. Introduction to Linux

- **What is Linux?:** Linux is a powerful and versatile open-source operating system based on the Unix architecture. It was created by Linus Torvalds in 1991 and has since grown into a widely-used platform for both personal and professional computing.
- **Open Source Nature:** One of the defining characteristics of Linux is that its source code is freely available for anyone to view, modify, and distribute. This has led to a collaborative environment where developers worldwide contribute to its development.
- **Kernel and Distributions:** Linux is composed of a kernel, which is the core component of the OS, and various distributions (distros) that bundle the kernel with software and package management systems. Popular distributions include Ubuntu, Fedora, Debian, and CentOS.
- **Linux in Different Environments:** Linux is used in a variety of environments, including desktops, servers, mobile devices, and embedded systems. Its flexibility allows it to run on a wide range of hardware, from supercomputers to small IoT devices.

2. Overview of the vi Editor

The vi (Visual Editor) is a powerful text editor available on almost all Unix-like operating systems, including Linux. It's known for its efficiency and versatility, particularly in environments where only a terminal interface is available. Here is a detailed look at the vi editor and its commands, presented in informative points.

1. Basics of vi Editor

- **Launching vi:** To start vi, type `vi filename` in the terminal. If `filename` does not exist, vi will create it.
- **Modes in vi:**
 - **Normal Mode:** The default mode where you can navigate and manipulate text.
 - **Insert Mode:** Used for inserting text. Enter by pressing `i`, `a`, or `o`.
 - **Command Mode:** Enter by typing `:` in Normal Mode for commands like saving, quitting, etc.
 - **Visual Mode:** Used to highlight and manipulate blocks of text.

2. Basic Commands for Running a C File

To work with C files in the vi editor, you only need a few basic commands to edit, save, and compile the file. Here's a simplified guide:

- **Open a File:** `vi filename.c`
 - Launches `vi` and opens the file named `filename.c`. If it doesn't exist, `vi` will create it.
- **Insert Mode:**
 - `i`: Enter Insert Mode before the cursor position.
 - `I`: Enter Insert Mode at the beginning of the line.
 - `a`: Enter Insert Mode after the cursor position.
 - `A`: Enter Insert Mode at the end of the line.
 - `o`: Open a new line below the current line and enter Insert Mode.
 - `O`: Open a new line above the current line and enter Insert Mode.
- **Save and Exit:**
 - `:w`: Save the file without exiting.
 - `:w filename`: Save the file with a new name.
 - `:q`: Quit `vi` without saving.
 - `:wq` **or** `ZZ`: Save the file and quit `vi`.
 - `:q!`: Quit without saving changes.

Implementation

Writing and Running a basic "Hello, World!" program in C using the terminal on a Linux system.

1. `cd ~/project`

2. `vi hello.c`

/* Save and Exit vi:

- Press Esc to exit Insert Mode.
- Type `:wq` and press Enter to save the file and quit `vi`.

***/**

3. `gcc hello.c -o hello`

4. `./hello`

```
#include <stdio.h>

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

~
~
~
~
:wq|
```

```
amit@Toshiba-Satellite-C850:~$ cd Downloads/
amit@Toshiba-Satellite-C850:~/Downloads$ vi hello.c
amit@Toshiba-Satellite-C850:~/Downloads$ gcc hello.c -o hello
amit@Toshiba-Satellite-C850:~/Downloads$ ./hello
Hello, World!
amit@Toshiba-Satellite-C850:~/Downloads$ |
```

Lab Exercise - 2

❖ AIM :: WAP in C to implement basic operations in different functions on Linux using vi-Editor

Source_Code ::

```
#include <stdio.h>

// Function to find the greatest number among three numbers
int findGreatest(int a, int b, int c)
{
    if (a > b && a > c) {
        return a;
    } else if (b > c) {
        return b;
    } else {
        return c;
    }
}

// Function to check if a number is even or odd
void evenOdd(int num)
{
    if (num % 2 == 0) {
        printf("%d is Even\n", num);
    } else {
        printf("%d is Odd\n", num);
    }
}
```

// Function to check if a number is prime

void checkPrime(int num)

```
{  
  
    int i, flag = 0;  
  
    if (num <= 1) {  
        printf("%d is not a Prime number\n", num);  
        return;  
    }  
  
    for (i = 2; i <= num / 2; ++i) {  
        if (num % i == 0) {  
            flag = 1;  
            break;  
        }  
    }  
  
    if (flag == 0) {  
        printf("%d is a Prime number\n", num);  
    } else {  
        printf("%d is not a Prime number\n", num);  
    }  
  
}
```

// Function to calculate the average of three numbers

double calculateAverage(int a, int b, int c) { return (a + b + c) / 3.0; }

int main()

```
{  
  
    printf("\n5C6 - Amit Singhal (11614802722)\n");  
  
    int num1, num2, num3;  
  
    int choice;  
  
    printf("\nChoose an operation:\n");  
  
    printf("1. Find Greatest of Three Numbers\n");  
  
    printf("2. Check Even or Odd\n");
```



```
printf("3. Check Prime Number\n");

printf("4. Calculate Average of Three Numbers\n");

printf("5. Exit\n");

while (1) {

    printf("\nEnter your choice: ");

    scanf("%d", &choice);

    switch (choice) {

        case 1:

            printf("\nEnter three numbers: ");

            scanf("%d %d %d", &num1, &num2, &num3);

            printf("Greatest Number: %d\n", findGreatest(num1, num2, num3));

            break;

        case 2:

            printf("\nEnter a number: ");

            scanf("%d", &num1);

            evenOdd(num1);

            break;

        case 3:

            printf("\nEnter a number: ");

            scanf("%d", &num1);

            checkPrime(num1);

            break;

        case 4:

            printf("\nEnter three numbers: ");

            scanf("%d %d %d", &num1, &num2, &num3);

            printf("Average: %.2f\n", calculateAverage(num1, num2, num3));

            break;

        case 5:

            printf("\n");

            return 0;

        default:
```

```
        printf("\nInvalid choice! Please choose again.\n");
    }
}

return 0;
}
```

Output ::

```
amit@Toshiba-Satellite-C850:~$ cd Desktop/Code/
amit@Toshiba-Satellite-C850:~/Desktop/Code$ vi basic_operations.c
amit@Toshiba-Satellite-C850:~/Desktop/Code$ gcc basic_operations.c -o basic_operations
amit@Toshiba-Satellite-C850:~/Desktop/Code$ ./basic_operations
```

5C6 - Amit Singhal (11614802722)

Choose an operation:

1. Find Greatest of Three Numbers
2. Check Even or Odd
3. Check Prime Number
4. Calculate Average of Three Numbers
5. Exit

Enter your choice: 1

Enter three numbers: 105 116 122

Greatest Number: 122

Enter your choice: 2

Enter a number: 13345

13345 is Odd

Enter your choice: 3

Enter a number: 5456527

5456527 is not a Prime number

Enter your choice: 4

Enter three numbers: 2234 4523 4355

Average: 3704.00

Enter your choice: 5

```
amit@Toshiba-Satellite-C850:~/Desktop/Code$ |
```

Lab Exercise – 2.2

- ❖ AIM :: WAP in C to implement basic operations in different functions on Linux using vi-Editor.

Source_Code ::

```
#include <stdbool.h>
#include <stdio.h>
#include <string.h>

// Function to print the Fibonacci series up to n terms
void fibonacci(int n)
{
    int first = 0, second = 1, next;

    if (n <= 0) {
        printf("Please enter a positive integer.\n");
        return;
    }

    printf("Fibonacci Series: ");
    for (int i = 1; i <= n; i++) {
        if (i == 1) {
            printf("%d ", first);
            continue;
        }
        if (i == 2) {
            printf("%d ", second);
```

```
        continue;
    }
    next = first + second;
    first = second;
    second = next;
    printf("%d ", next);
}
printf("\n");
}
```

// Function to calculate the factorial of a number

```
int factorial(int n)
{
    if (n == 0) {
        return 1;
    }
    return n * factorial(n - 1);
}
```

// Function to calculate the sum of digits of a number

```
int digitsSum(int num)
{
    int sum = 0;
    while (num != 0) {
        sum += num % 10;
        num /= 10;
    }
    return sum;
}
```

// Function to check if a string is a palindrome

```
bool isPalindrome(char str[])
```

```
{  
    int length = strlen(str);  
    int start = 0;  
    int end = length - 1;  
  
    while (start < end) {  
        if (str[start] != str[end]) {  
            return false;  
        }  
        start++;  
        end--;  
    }  
    return true;  
}
```

// Function to count the occurrences of a character in a string

```
int countChar(char* str, char ch)  
{  
    int count = 0;  
    for (int i = 0; str[i] != '\0'; i++) {  
        if (str[i] == ch) {  
            count++;  
        }  
    }  
    return count;  
}
```

```
int main()
```

```
{  
  
    int choice, num1, num2, num3;  
    char str[100], ch;
```

```
printf("\n5C6 - Amit Singhal (11614802722)\n");
```

```
// Display the menu
```

```
printf("\nMenu:\n");
```

```
printf("1. Print Fibonacci Series\n");
```

```
printf("2. Calculate Factorial\n");
```

```
printf("3. Calculate Sum of Digits\n");
```

```
printf("4. Check Palindrome\n");
```

```
printf("5. Count Character Occurrences\n");
```

```
printf("6. Exit\n");
```

```
while (1) {
```

```
    printf("\nEnter your choice (1-6): ");
```

```
    scanf("%d", &choice);
```

```
    switch (choice) {
```

```
        case 1:
```

```
            printf("\nEnter the number of terms for Fibonacci series: ");
```

```
            scanf("%d", &num1);
```

```
            fibonacci(num1);
```

```
            break;
```

```
        case 2:
```

```
            printf("\nEnter a number to calculate its factorial: ");
```

```
            scanf("%d", &num1);
```

```
            printf("Factorial: %d\n", factorial(num1));
```

```
            break;
```

```
        case 3:
```

```
            printf("\nEnter a number to calculate the sum of its digits: ");
```

```
            scanf("%d", &num1);
```

```
            printf("Sum of Digits: %d\n", digitsSum(num1));
```

break;

case 4:

printf("Enter a string to check if it is a palindrome: ");

scanf("%s", str);

if (isPalindrome(str)) {

printf("%s is a Palindrome\n", str);

} else {

printf("%s is not a Palindrome\n", str);

}

break;

case 5:

printf("\nEnter a string: ");

scanf("%s", str);

printf("Enter a character to count its occurrences: ");

scanf(" %c", &ch);

printf("Count of '%c': %d\n", ch, countChar(str, ch));

break;

case 6:

printf("\nExiting the program. Have a great day!\n");

return 0;

default:

printf(

"\nInvalid choice! Please select a number between 1 and 6.\n");

}

}

return 0;

}

Output ::

```
amit@Toshiba-Satellite-C850:~/Downloads/OS$ vi basic_operations_2.c
amit@Toshiba-Satellite-C850:~/Downloads/OS$ gcc basic_operations_2.c -o prg_2
amit@Toshiba-Satellite-C850:~/Downloads/OS$ ./prg_2

5C6 - Amit Singhal (11614802722)

Menu:
1. Print Fibonacci Series
2. Calculate Factorial
3. Calculate Sum of Digits
4. Check Palindrome
5. Count Character Occurrences
6. Exit

Enter your choice (1-6): 1

Enter the number of terms for Fibonacci series: 9
Fibonacci Series: 0 1 1 2 3 5 8 13 21

Enter your choice (1-6): 12

Invalid choice! Please select a number between 1 and 6.

Enter your choice (1-6): 2

Enter a number to calculate its factorial: 12
Factorial: 479001600

Enter your choice (1-6): 3

Enter a number to calculate the sum of its digits: 35544355
Sum of Digits: 34

Enter your choice (1-6): 4
Enter a string to check if it is a palindrome: madam
madam is a Palindrome

Enter your choice (1-6): 5

Enter a string: helloworld
Enter a character to count its occurrences: l
Count of 'l': 3

Enter your choice (1-6): 6

Exiting the program. Have a great day!
amit@Toshiba-Satellite-C850:~/Downloads/OS$ |
```


Lab Exercise - 3

❖ AIM :: WAP in C to implement CPU scheduling for first come first serve.

Source_Code ::

```
#include <stdio.h>

#include <stdlib.h>

#define MAX 100

typedef struct
{
    int pid;
    int burst_time;
    int waiting_time;
    int turnaround_time;
} Process;

void print_table(Process p[], int n);
void print_gantt_chart(Process p[], int n);

int main()
{
    Process p[MAX];
    int i, j, n;
    int sum_waiting_time = 0, sum_turnaround_time = 0;
```

```
printf("\n5C6 - Amit Singhal (11614802722) \n");
```

```
printf("\nEnter total number of processes: ");
```

```
scanf("%d", &n);
```

```
printf("\nEnter burst time for each process:\n");
```

```
for (i = 0; i < n; i++)
```

```
{
```

```
    p[i].pid = i + 1;
```

```
    printf("P[%d]: ", i + 1);
```

```
    scanf("%d", &p[i].burst_time);
```

```
    p[i].waiting_time = 0;
```

```
    p[i].turnaround_time = 0;
```

```
}
```

```
p[0].turnaround_time = p[0].burst_time;
```

```
for (i = 1; i < n; i++)
```

```
{
```

```
    p[i].waiting_time = p[i - 1].waiting_time + p[i - 1].burst_time;
```

```
    p[i].turnaround_time = p[i].waiting_time + p[i].burst_time;
```

```
}
```

```
for (i = 0; i < n; i++)
```

```
{
```

```
    sum_waiting_time += p[i].waiting_time;
```

```
    sum_turnaround_time += p[i].turnaround_time;
```

```
}
```

```
puts("");
```

```
print_table(p, n);
```

```
printf("\nTotal Waiting Time: %d\n", sum_waiting_time);
```

```
printf("Average Waiting Time: %.2lf\n", (double)sum_waiting_time / (double)n);
```

```
printf("Total Turnaround Time: %d\n", sum_turnaround_time);
```

```
printf("Average Turnaround Time: %.2lf\n", (double)sum_turnaround_time /  
(double)n);
```

```
puts("\nGANTT CHART\n");
```

```
print_gantt_chart(p, n);
```

```
return 0;
```

```
}
```

```
void print_table(Process p[], int n)
```

```
{
```

```
int i;
```

```
puts("+-----+-----+-----+-----+");
```

```
puts("| PID | Burst Time | Waiting Time | Turnaround Time |");
```

```
puts("+-----+-----+-----+-----+");
```

```
for (i = 0; i < n; i++)
```

```
{
```

```
printf("| %3d | %10d | %12d | %15d \n", p[i].pid, p[i].burst_time, p[i].waiting_time,  
p[i].turnaround_time);
```

```

    puts("+-----+-----+-----+-----+");
}
}

```

```

void print_gantt_chart(Process p[], int n)

```

```

{

```

```

    int i, j;

```

```

    // Top border of the Gantt chart

```

```

    puts("+----+----+----+----+----+----+");

```

```

    // Process IDs

```

```

    puts("| P1 | P2 | P3 | P4 | P5 | P6 |");

```

```

    // Bottom border of the Gantt chart

```

```

    puts("+----+----+----+----+----+----+");

```

```

    for (i = 0; i < n; i++)

```

```

    {

```

```

        printf("| %d ", p[i].turnaround_time);

```

```

        if (p[i].turnaround_time > 9)

```

```

            printf("\b"); // Remove 1 space if the number has 2 digits

```

```

    }

```

```

    printf("\n");

```

```

    puts("+----+----+----+----+----+----+");

```

```

    printf("\n");

```

```

}

```

Output ::

```
singhal-amit@singhal-amit-ThinkPad-T430:~/Downloads/_LAB_Work/OS/Code$ vi prg_3_fcfs.c
singhal-amit@singhal-amit-ThinkPad-T430:~/Downloads/_LAB_Work/OS/Code$ gcc prg_3_fcfs.c -o a
singhal-amit@singhal-amit-ThinkPad-T430:~/Downloads/_LAB_Work/OS/Code$ ./a
```

5C6 - Amit Singhal (11614802722)

Enter total number of processes: 6

Enter burst time for each process:

P[1]: 5

P[2]: 3

P[3]: 9

P[4]: 8

P[5]: 4

P[6]: 7

PID	Burst Time	Waiting Time	Turnaround Time
1	5	0	5
2	3	5	8
3	9	8	17
4	8	17	25
5	4	25	29
6	7	29	36

Total Waiting Time: 84

Average Waiting Time: 14.00

Total Turnaround Time: 120

Average Turnaround Time: 20.00

GANTT CHART

P1	P2	P3	P4	P5	P6
5	8	17	25	29	36