Program Set 8

Dates: 09.01.2023 – 13.01.2023

Introduction to Ubuntu/Unix/Linux/Redhat:

1. Unix is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, whose development started in 1969 at the Bell Labs research centre by Ken Thompson, Dennis Ritchie, and others.
2. Almost the entire operating system is written in the C programming language, which allows Unix to operate on numerous platforms.
3. Unix allows direct communication with the computer via a terminal, hence being very interactive and giving the user direct control over the computer resources.
4. Unix is not free. So we use Ubuntu/Redhat etc.
5. We use unix-like os which are open source – what does that mean?
    a. Open source software is code that is designed to be publicly accessible—anyone can see, modify, and distribute the code as they see fit. Open source software is developed in a decentralized and collaborative way, relying on peer review and community production. – You can directly modify what you need.
    b. Example: Mozilla firefox, VLC, Libreoffice
6. Direct communication – commands for everything – via terminal.

Things you need to know about to write C programs in Ubuntu:

1. Terminal: The terminal (also known as console) is an application in which you can execute text commands directly.

2. vi/vim editor: The default editor that comes with the UNIX operating system is called vi (visual editor). Using vi editor, we can edit an existing file or create a new file from scratch. we can also use this editor to just read a text file.
3. GCC: GCC stands for GNU Compiler Collections which is used to compile mainly C and C++ language.

How to open terminal and use it:

1. Application → Terminal
2. Common commands in Ubuntu: (after you write a command please press **enter**)
    a. pwd: This command refers to the present working directory in which you are operating.
    b. dir: The dir command is used to print (on the terminal) all the available directories in the present working directory
    c. ls: This command is used to list down all the directories and files inside the present working directory
    d. cd: you can change the directories in the terminal. Try **cd Desktop**
    e. mkdir: It will make a directory in your pwd; for example, the command: **mkdir new** will make the directory "new" in pwd.
    f. rm: This remove command is used to remove the specific file from a directory; (later)
    g. cp: The cp command will help you to copy any file or folder to any directory (later)
    h. man: The man command will help you to get the complete user manual of any specific command. (later)

Write and run a C code using the Terminal

1. Navigate to Desktop:
    a. Command: cd Desktop
2. Create a folder with your name:
    a. mkdir yourname
3. Change your pwd to this new directory/folder:
    a. Command: cd yourname
4. Create a file with the name prog1.c using vi editor:
    a. Command: vi prog1.c
5. Switch to insert mode:
    a. Command: press i
6. Write your code here.
7. Save your file:
    a. Command: press the **ESC** button (this switches you to command mode)
    b. Command: Press **:wq** (Save and quit/exit vi. Takes you back to terminal)
8. Compile your code:
    a. Command: **gcc prog1.c** (in case of any errors you will be notified here)
9. To run your code:
    a. Command: **./a.out**

Programs:

1. Write a C program to declare an integer array of size 10 and take user inputs to populate the array. Print the sum of the array elements.
2. Ten distinct integers are entered from the keyboard into an array. Also, the number to be searched (key) is entered through the keyboard by the user. Write a program to find if the key is present in the array or not.
    a. Use Linear search
    b. Use binary search
3. Write the above program for an array which can contain the key multiple times, and display the number of times it appears in the array along with its indices.
4. Sort an integer array of size 10 using Bubble sort.
    a. In ascending order
    b. In descending order
5. Write a C program to display the contents of an user input integer array of size 6, using a function. Pass each element of the array to function using
    a. Call by value
    b. Call by reference
6. Write a C program to calculate the sum of array elements in a float type array of size 10, by passing entire array to a function (do not pass individual elements).
7. Access the elements of an user input integer array of size 15 using pointer notation and print them.
8. Find the smallest number in an integer array of size 8 using pointer notation.
9. Write a program to declare an integer 2D array of 4 rows and 5 columns. Take user inputs to populate it. Print the elements of the array. Also print the sum and product of all the elements of the array. Use index notation.
10. Write the above program using pointer notation.

```c
#include <stdio.h>
int main() {
    int A[5], B[5], i;
    //Using index notation to access array elements
    printf("Enter elements: ");
    for (i = 0; i < 5; i++)
        scanf("%d", &B[i]);
    printf("You entered: \n");
    for (i = 0; i < 5; i++)
        printf("%d\n", B[i]);
    printf("\n----------------------\n")        ;
    //Using pointer notation to access array elements
    printf("Enter elements: ");
    for (i = 0; i < 5; i++)
        scanf("%d", (A + i)); //note the use of base address of A.
    //Since A indicates the base address, so & is not required.
    printf("You entered: \n");
    for (i = 0; i < 5; i++)
        printf("%d\n", *(A + i));
    return 0;
}
```