

NAME: Razaq Hammed Abiola

MATRIC NO: 230927

LEVEL: 200L

SOLUTION TO THE ASSIGNMENT

1. Write briefly on Unix operating system especially linux flavor
 2. Write a short note on software functional requirements
 3. Why does unix often prefer at some points?
 4. Why does unix being referred to as a Scientist OS?
 5. What type of programming language is C?
 6. Give the detailed structure of a complete C programming language
 7. How can I create a C programming file on the OS
 8. Write a program in C to say Hello World to your lecturer
 9. Write a C program to accept the names of 10 students
 10. Write a C program to count 1-N
 11. Write a C programming language that reverses a sentence
 12. Write a program in C to solve a simple quadratic equation ($x^2+y+c=0$). Your program you allow the use of any number as the coefficient of any of the variables.
- Note: The execution should be done on any of the unix flavor. Submit this project on both the LMS and your Github account.

Solution

1. **brief on Unix operating system especially linux flavor**

Unix: this is an operating system analogous to DOS and windows that supports multiple concurrent users. Or

A modular operating system made up of a number of essential components including the kernel, shell, file system and core set of utilities or programs.

Unix is made up by three parts

The kernel

The shell and

The programs

The kernel :- This is the hub of the operating system , it allocates time and memory to the programs, handles the file stored and communicates in response to system calls.

The shell :- This act as an interface between user and the kernel i.e it interpret the command user types in and arranged for them to be carried out.

The program :- This is file or process

File: Collection of data

Process: executing program by pid (process identifier)

The properties of Unix

- i. Ability to run on many different system
- ii. Simplicity
- iii. Portability
- iv. Multitasking capacity
- v. Multiuser capacity
- vi. Hierarchical file system

Examples of Unix

Mac OS

Solaris

Aix

Further on Linux flavor

10 most stable Linux distros

- i. Arch Linux
- ii. Debian
- iii. Fedora
- iv. Linux mint
- v. Manjaro
- vi. Tails
- vii. OpenSuse
- viii. Ubuntu
- ix. Android
- x. CentOS
- xi. Elementary OS
- xii. Kali Linux

Three different categories of Linux flavor

Security-focused

User focused and

Unique

Examples of Linux flavor

Debian

Ubuntu

Fedora

Suse

Mandriva

Debian:- is a Linux based operating system for a wide range of devices including, desktop and server.

Ubuntu:- this is a Linux distribution based on Debian composed mostly of free and open source software with their edition

Desktop

Robots and

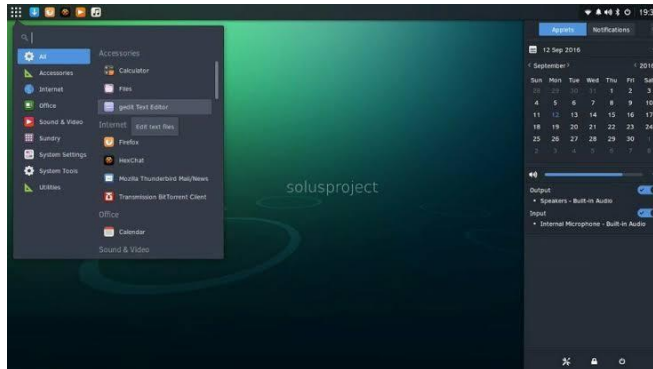
Core

Fedora:- is a Linux distribution developed by the fedora project, contain software distributed under various free and open source technologies

Default user interface: GNOME shell, Bash

Suse:- Software-undsystem entwicklung

Suse this is German based multinational open source software company by Roland Dyroff



2. Write a short note on software functional requirements

What are functional requirements:- this is a featured or functions that developers must implement to enable users to accomplish thier tasks

Types of functional requirements

- i. The regulation
- ii. The certification
- iii. The report
- iv. The function
- v. Audit
- vi. External interface
- vii. Data management
- viii. Legal and regulatory requirement
- ix. Security

The Merits

- i. It allows you to determine the application has all the functionalities specified in the functional requirements.
- ii. It allows you to identify missing requirements
- iii. The most cost effective time to correct errors during the functional requirements gathering stage
- iv. You can express functional requirements in use case forms

3. Why does unix often prefer at some points?

Unix is often preferred at some point because

- i. It allows you to open multiple programs, and all program execute in parallel using multiprocessor technology
- ii. It quick and error free
- iii. Each programs runs it's own server and needed with it's own username on the system
- iv. Portability i.e ability to run on different systems
- v. Multiuser and multitasking capabilities
- vi. Extension library of software
- vii. Hierarchical file system.
- viii.

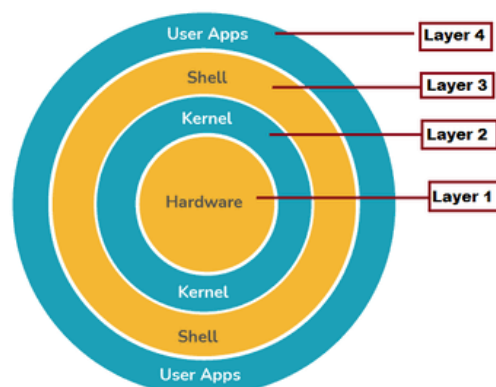
4. Why does unix being referred to as a Scientist OS?

Unix is being referred to as scientific operating system because its are freaking alternative to monolithic tools IDEs and languages like Java

Because of it's

- i. Support to many users and multitasking
- ii. Being free and customization
- iii. Portable being created using the language i.e highly portable and easy to use
- iv. Less memory usage:- handle virtual memory more efficiently
- v. Uniformity:-all data in Unix is stored on files, device, and terminals
- vi. Security:- it provides save and secure platform for multi users.
- vii. It's ideal for web hosting
- viii. Modular:- Unix OS divide a bigger problems to smaller ones
- ix. Easy remove of files due to hierarchical file system
- x. Better performance and stable database access

Structure of Unix operating system layers



5. What type of programming language is C?

C is a structured ,procedural programming language that has been widely used to create both operating system and applications such as

Oracle database

Windows

Git

Python interpreter and games.

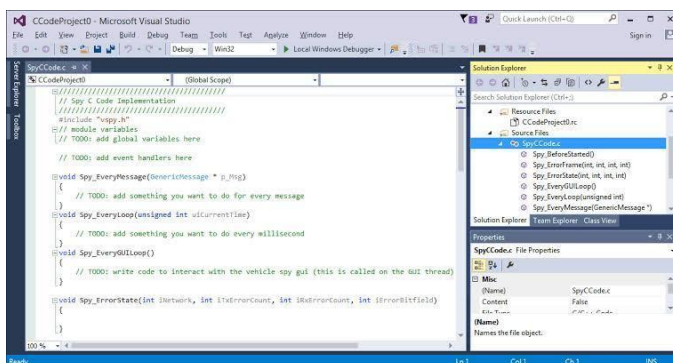
C was created in the 1970s by Dennis M. Richie

C data type

- i. Integer
- ii. Float
- iii. Double
- iv. Chat

Type	Size (bits)	Size (bytes)	Range
char	8	1	-128 to 127
unsigned char	8	1	0 to 255
int	16	2	-2^{15} to $2^{15}-1$
unsigned int	16	2	0 to $2^{16}-1$
short int	8	1	-128 to 127
unsigned short int	8	1	0 to 255
long int	32	4	-2^{31} to $2^{31}-1$
unsigned long int	32	4	0 to $2^{32}-1$
float	32	4	3.4E-38 to 3.4E+38
double	64	8	1.7E-308 to 1.7E+308
long double	80	10	3.4E-4932 to 1.1E+4932

C program code area interface



6. Give the detailed structure of a complete C programming language

A C program is divided into six sections:

- I. Documentation,
- II. Link,
- III. Definition,
- IV. Global Declaration,
- V. Main() Function, and
- VI. Subprograms.

Documentation

C program, single-line comments can be written using two forward slashes (i.e., //, and we can create multi-line comments using /* */). Here, we've used multi-line comments.

```
/** multiple  
line comments  
*/
```

Link

A header file is a file that consists of C declarations that can be used between different files. It helps us in using others' code in our files. A copy of these header files is inserted into your code before the compilation.

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

Definition

A preprocessor directive in C is any statement that begins with the naira sign "#" symbol. The #define is a preprocessor compiler directive used to create constants. In simple terms, #define basically allows the macro definition, which allows the use of constants in our code.

#define is typically used to make a source program easy to edit and compile in different execution environments.

The define statement **does not** ends with a semicolon (";").

Global Declaration

This section includes all global variables, function declarations, and static variables. The variables declared in this section can be used anywhere in the program. They're accessible to all the functions of the program. Hence, they are called global variables.

```
int age(int current);
```

We've declared our age function, which takes one integer argument and returns an integer.

Main() Function

In the structure of a C program, this section contains the main function of the code. The compiler starts execution from the main() function. It can use global variables, static variables, inbuilt functions, and user-defined functions. The return type of the main() function can be void and also not necessarily int.

```
int main(void){
int current = 2022;
printf("Age: %d", age(current));
return 0;
}
```

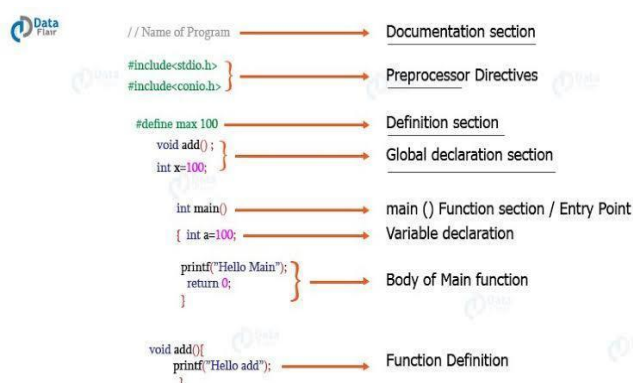
Here, we've declared a variable named current and assigned the value as 2021. Then we've called the printf() function, which calls the age() function, which takes only one parameter.

Subprograms

This comprises the user-defined functions named in the main() function. User-defined functions are generally written after the main() function irrespective of their order. When the user-defined function is called from the main() function, the control of the program shifts to the called function, and when it encounters a return statement, it returns to the main() function. In this case, we've defined the age() function, which takes one parameter, (i.e the current year).

```
int age (int curent){
return current - BORN;
}
```

This function is called in the main function. It returns an integer to the main function.



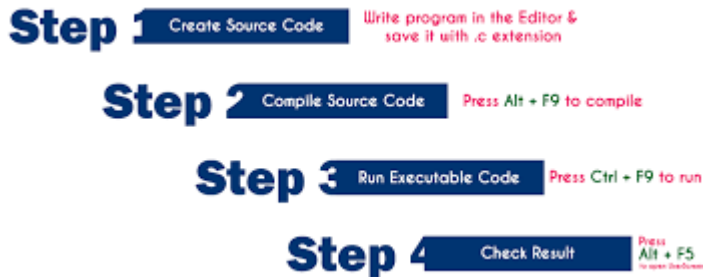
```
untitled X
1 #include <stdio.h>
2 int main()
3 {
4     printf("Hello, World !");
5     return 0;
6 }
```


7. How to create c programming file on the operating system

In windows operating systems

We need to install **Turbo C software**, or download **Dev Cpp compiler**

- i. Create source code
- ii. Compile source code
- iii. Run executable code
- iv. Check results



Step 1:

create a source code:- Source is a file C programming instructions in high level language

to create source any text editor can be use in written the program instructions

Steps in creating source code file in windows Operating system

Click on **Start** button

Select **Run**

Type **cmd** and press enter

Type **cd c:\TC\bin** in the command prompt and press enter

Type **TC** press **Enter**

Click on **File -> New** in C editor Window

Type the **program**

Save it as **FileName,c** (use shortcut key F2 to save)

Step 2: Compile Source Code (Alt + F9)

The compilation is the process of converting high-level language instructions into low-level language instructions. We use the shortcut key **Alt + F9** to compile a C program in **Turbo C**.

Whenever we press **Alt + F9**, the source file is going to be submitted to the Compiler. On receiving a source file, the compiler first checks for the Errors. If

there are any Errors then compiler returns List of Errors, if there are no errors then the source code is converted into **object code** and stores it as a file with **.obj** extension. Then the object code is given to the **Linker**. The Linker combines both the **object code** and specified **header file** code and generates an **Executable file** with a **.exe** extension.

Step 3: Executing / Running Executable File (Ctrl + F9)

After completing compilation successfully, an executable file is created with a **.exe** extension. The processor can understand this **.exe** file content so that it can perform the task specified in the source file.

We use a shortcut key **Ctrl + F9** to run a C program. Whenever we press **Ctrl + F9**, the **.exe** file is submitted to the **CPU**. On receiving **.exe** file, **CPU** performs the task according to the instruction written in the file. The result generated from the execution is placed in a window called **User Screen**.

Step 4: Check Result (Alt + F5)

After running the program, the result is placed into **User Screen**. Just we need to open the User Screen to check the result of the program execution. We use the shortcut key **Alt + F5** to open the User Screen and check the result.

Must know:-

1. C program file (**Source file**) must save with **.c extension**.
The compiler converts complete program at a time from high-level language to low-level language.
2. Input to the compiler is **.c** file and output from the compiler is **.exe** file, but it also generates **.obj** file in this process.
3. The compiler converts the file only if there are no errors in the source code.
4. **CPU** places the result in **User Screen** window.

Overall Process

1. Type the program in C editor and save with **.c extension** (Press **F2** to save).
2. Press **Alt + F9** to compile the program.

3. If there are errors, correct the errors and recompile the program.
4. If there are no errors, then press **Ctrl + F9** to execute/run the program.
5. Press **Alt + F5** to open User Screen and check the result.

In Unix operating systems

Linux

1. Open terminal
2. Use the vim editor
3. Open file using,
4. vim file.c (file name can be anything but it should end with dot c extension) command.

5. To Edit the file:
6. Press i to go to insert mode.
7. Type your program.
8. To save the file:

Press Esc button and then type :wq. It will save the file.

To compile the program:

9. Type, gcc file.c
10. To Run the program: Type,. ./a.out

8. Write a C program to say “Hello world” to your lecturer

```
9. #include <stdio.h>
10. int main()
11. {
12.     /* program to say hello Word to your lecture
13.
14.     */
15.     printf("Hello World Dr. Ayinla ");
16.     return 0;
}
```

