NAME: NWABUIFE DIVINE OLUCHUKWU

MATRIC NO.: 222484  
DEPARTMENT: COMPUTER SCIENCE

**CSC 235: Assignment on Unix and C language**

**Question1**

Unix Operating System

Unix Operating System is a multiuser (as it can be used as several people at the same time) and multitasking (as it can run multiple programs at the same time) operating system, designed for flexibility and adaptability. It is a set of programs which serves as a link between the computer and the user It's development began at 1969, at the Bell Labs Research center, and it was first developed by Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna. The operating system has been under development ever since.

The Unix Operating System is majorly written in C programming language, and this enables its operation on numerous platforms. Unix Operating System exists in different versions/flavours.These versions of Unix are united by a main concept with four basics. These four basics are: Kernel, Shell, Commands and Utilities, Files and Directories. Some versions/flavours of the Unix Operating System are: Sun Solaris, GNU/Linux, and MacOS X.

**Linux Flavour:**

Linux is an open source operating system which is Unix-like. It is a community-developed operating system (OS) for computers, servers, mainframes, mobile devices and embedded devices.It is one of the most widely supported operating systems as it is supported on almost every major computer platform.

Linux can be used in:

* Server OS
* Desktop OS
* Headless server OS
* Embedded device or appliance OS
* Network OS
* Software development OS
* Cloud OS

Linux is available in hundreds of versions known as distributions or distros, which are usually distinguished by their specific goal, philosophy, function or target market.

Examples of Linux distribution are:

* Ubuntu Linux
* Linux Mint
* Kali Linux
* Debian Linux
* TAILS
* SUSE Linux, etc.

Linux is made up of several components, including:

* Bootloader
* Kernel
* Init system
* Daemons
* Graphical server
* Desktop environment
* Applications

**Question2**

Software Functional Requirements

Software Functional Requirements are the features or functions a developer must implement in a product to enable users to successfully carry out their tasks. It is a function of a system or the component of a system where a function is described as a specification of behavior between inputs and outputs. Software Functional Requirements can involve calculations manipulation and processing of data, technical details and other functions that define what a system should accomplish.

Functional Requirements are supported by non-functional requirements which are also known as from quality requirements.

**Question3**

Why Is Unix Preferred at Some Points?

One of the reasons why Unix is preferred at some points because

of its portability, as it is written in C, which is a high level language, and so makes its installation on new computing systems easy. Applications written to run on one Unix system, can most likely run on any Unix system irrespective of the hardware

Another reason is that Unix provides a range of tools that can be combined and manipulated to perform a wide variety of jobs that enables users of the system carry out sophisticated tasks without needing to write programs in a programming language.

Again, Unix comes on various versions which are largely compatible.

**Question4**

Why Is Unix Referred To As A Scientific OS?

Unix is referred to as a scientific operating system as it is widely used for scientific operations like data science due to the advantages it provides with analysing data.

**Question5**

What Type of Programming Language is C

C is a structured and procedural general all-purpose programming language. It is a structured programming language because it has a standard structure and pattern in which it is written. It is referred to as a procedural language as it follows specified steps in solving a problem. Also, C programming language is considered as a middle-level language because it supports the feature of both low-level and high-level languages.

**Question6**

Give detailed the structure of a complete C program

A complete C program is made up of certain basic components structured in a particular way.

Below is the structure of a complete C program:

*Header* - #stdio h

*main()* - int main()

{

*Variable Declaration* - char word [] = "Change"

*Body* - printf ("%s",word);

*Return* - return 0;

}

Components of a complete C program

1. **Header Files Inclusion**:

A header file is a file with extension .h that contains C function declarations and macro definitions to be shared between several source files.

Examples of C Header files:

*stddef.h – Defines several useful types and macros.*

*stdint.h – Defines exact width integer types.*

*stdio.h – Defines core input and output functions*

*stdlib.h – Defines numeric conversion functions, pseudo-random network generator, memory allocation*

*string.h – Defines string handling functions*

*math.h – Defines common mathematical functions*

1. **Main Method Declaration**: The next part of a complete C program is the part where main() function is declared.

Below is the syntax for declaring main() function

*int main()*

*{}*

1. **Variable Declaration**: This is the next part of a C program and it refers to the variables that are to be used in the function. In the C program, a variable cannot be used unless it is declared, and the variables are be declared before any operation function.

Below is an example of a valid syntax for variables declaration:

int main()

{

char word [];

}

1. **Body**: The body is the next part of a C program. The body of function in the C program, refers to the operations (manipulations, searching, sorting, printing) that are performed in the functions. It can be anything like printing, sorting, manipulations, searching, etc.

int main()

{

char word

printf("%s", word);

.

1. **Return Statement**: This is the last part of any C program. It refers to the returning values from a function. This return statement and return is dependent on the return type of the function. For example, if the return type is void, then there will be no return statement. In any other case, there will be a return statement and the return value will be of the type of the specified deferred return type.

Below is the syntax for the return Statement:

int main()

{

char word

printf("%s", word);

return 0;

}

**Question7**

Creating a C Program File in Linux

To create and run a C program on Linux, one needs to have a compiler installed.

How to create a C program file in Linux OS and compile it using GCC compiler.

GCC can be installed using the *apt package manager* .

Command for installation: **$ sudo apt install gcc**

After installation, create a file with an extension of **.*c*** e.g *task.c* using the nano editor. The extension tells the computer that this file contains a C program.

Command for file creation: **$ nano task.c**

Write your C program in the editor then press CTRL+S to save. To exit the editor, press CTRL+X.

To verify your file creation lists down its contents, enter the command *ls*

Command for listing content of files: **$ls**

Now, after successfully confirmingthe creation of your file and its contents, compile the file using the GCC compiler, and run the command:

**$ gcc task.c -o task**

In the above command, the GCC compiler compiles the file which was created with the name of task.c and then checks whether it has an error or not. The “task” (binary file) will be generated in the same directory if there are errors. To execute the binary file use: **$ ./task**

The output of the C program will then be displayed.

Asides using the terminal, this process can also been executed using different compilers like Visual Studio that are GUI (Graphical User Interface) based. These compilers can be installed on Linux to compile and run C programs in Linux.