**NAME:** ADEAGBO PROMISE AYODEJI

**MATRIC NUMBER:** 222449

**DEPARTMENT:** COMPUTER SCIENCE

**COURSE CODE:** CSC 235

**LECTURER:** DR AYINLA

**1. Write briefly on Unix Operating System especially Linux Flavor**

Unix Operating System is a multi-user (i.e., several people can use a UNIX computer at the same time), multi-tasking (i.e., a user can run multiple programs at the same time) and multi-processing operating system. It is a command line interpreter which means one can perform tasks done on the Graphical User Interface using the Command Line Interface. It was developed by AT&T Bell Laboratories Research Center, USA in 1969 by Ken Thompson and Denis Ritchie. The entire UNIX operating system is written in the C programming language which allows UNIX to operate on numerous platforms, however earlier UNIX was written in Assembly language and originally spelled as UNICS(Uniplexed Information Computing System Service). There are various UNIX variants available today some of which include Solaris UNIX, IBM AIX, HP-UX, BSD (FreeBSD, OpenBSD, NetBSD), Linux (a flavor of UNIX which is freely available) etc.

The main concept that unites all the versions of UNIX is the following 4 features which include:

1. SHELL: this is the component that processes the users' requests. When a command is typed into the terminal, the shell interprets the command and calls the program the user wants.

2. KERNEL: this is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, file management and task scheduling.

3. FILES AND DIRECTORIES: all the data of UNIX is organized into files. All files are then organized into directories. The directories are then further organized into a structure called the FILESYSTEM.

4. COMMANDS AND UTILITIES: the UNIX operating system comes with built in commands and utilities which can be made use of in day-to-day activities such as df, find, mkdir, ls etc.

Pros and Cons of the UNIX operating system

1. Some drivers are not available for the software.

2. The User Interface is not so much attractive.

3. Some variants of UNIX requires most memory usage hence this is not suitable for a device with low level of memory.

4. UNIX has fewer users so most game companies do not make games for UNIX. UNIX has very fewer games available than other OS like windows and macOS.

LINUX

Linux is an open-source UNIX like Operating System based on the Linux kernel. It is one of the most popular platforms today due to several reasons such as:

1. it is free to download and use.

2. it is generally far less vulnerable to attacks.

LINUX FLAVORS

Linux has a number of different versions to suit any type of user. From new users to hard-core users, there are "FLAVORS" of Linux to match a user's need. These versions are called "**DISTRIBUTIONS('distros')**". Nearly every distribution of Linux can be downloaded for free and installed (on as many machines as possible). Popular Linux distributions include the following:

1. **Arch Linux**: this Linux distribution is popular amongst developers. It is an independently developed system. It is not recommended for beginners because every package has to be installed by the user itself and it also provides the user the freedom to customize the system to their preference.

2. **CentOS (Community Enterprise Operating System)**: one of the most used Linux distributions for enterprises and web servers. It is a free enterprise class OS and it based heavily on RedHat enterprise Distribution.

3. **GENTOO Linux**: It is a source-based distribution which means the user needs to configure the code on the system before they can install it. It is not for Linux beginners but it is fun for experienced users because it allows them to have a deeper understanding of the ins and outs of the Linux operating system.

4. **Linux Mint**: one of the most popular desktop distributions available out there. it launched in 2006 and is now considered with as fourth most used OS in the computing world. For a user who wants a fast and stable Linux desktop to perform day-to-day desktop tasks, gaming, watching videos etc., this is the highly recommended distribution to download.

5. **Ubuntu**: this is one of most popular desktop OS across the globe today. it is based on Debian Linux distribution and it is known for its desktop environment. It was specifically designed for beginners in Linux or those transitioning from Mac OS and Windows. Due to the user-friendliness and beautiful UI feature of Ubuntu, it is ideal for newcomers trying to wrap their head around Linux. Each Linux distribution is built for a specific purpose to meet the demands of its target users. Most of the distributions are available for free at their respective websites.

6. **Fedora**: it is regarded as one of the most user-friendly distributions due to its simplicity and ease of use for newcomers. It is a very powerful operating system well suited for laptops, servers etc. It is used for development and learning purposes by developers and students.

7. **Kali Linux**: it is a Debian-based Linux distribution recommended for Cybersecurity students and experts who wish to venture into penetration testing.

There are several Linux distributions available today, however some of the popular ones were those listed above.

**2.) Write a short note on software functional requirements**

Requirements which are related to the effective functional aspect of software falls into 2 categories:

1. Software Functional Requirements

2. Non-Functional Requirements

**Software Functional Requirements**: These are requirements that specify what a system or software should do. They are product features that focus on user needs. They are usually answer the question "WHAT THE SYSTEM MUST DO?". It describes the behaviors of the system under specific conditions e.g., "send verification email to a new customer who signs up on the system".

If a system does not meet a functional requirement, it will fail. This is because the system will not be able to achieve what it must do to operate effectively.

Functional requirements usually specify what the system must do in response to inputs and what it must provide as output. Functional requirements basically consist of both the product features and user requirements. Examples of functional requirements include the following:

1. What a system/software does when a user clicks on a certain button, where are they taken next?

2. Authentication of user whenever he/she logs into the system?

3. Steps that must be taken for a new user to create an account on a software/system?

4. What happens in the event of an attack on the system?

ADVANTAGES OF FUNCTIONAL REQUIREMENTS

1. It enables developers to plan the entire development process of a product easily

2. It helps to produce error free products and also offers clients the exact product they demanded for.

**3.) WHY IS UNIX OFTEN PREFERED AT SOME POINTS**

1. UNIX is open source this means the operating system is freely available to the community for use and customization.

2.) MULTITASKING: UNIX operating system allows the user to launch several programs or processes from the same terminal with one running in the foreground and the other in the background.

3.) UNIX is more stable and does not crash as often as other OS such as Windows. Therefore, it requires less administration and maintenance.

4.) SECURITY: UNIX has greater built-in security and permission features. UNIX provides a safe and secure environment for multiple users. Each user requires to be authenticated to access the operating system therefore files are protected from unauthorized use.

5.) UNIX possess much greater processing power thus allowing it handle high server loads than windows and UNIX machines seldom require reboots while Windows is constantly needing them.

**4.) Why is UNIX referred to a Scientist OS?**

Unix is often regarded as the operating system of data scientists.

This is due to the following reasons such as:

1. It runs much smoother and faster than other operating systems like Windows thus allowing the user to perform heavier computational task without any issues.

2. It has a powerful command line which can perform range of tasks from simple to complex ones in a small amount of time.

3. Most operations carried out by data scientists involves them having to interact with UNIX systems, hence they have to know how to use them to carry out activities.

**5. What type of programming language is C?**

C is a general purpose, procedural (instructions in C program are executed step by step) programming language which relates closely to the way machines work. It was created by Dennis Ritchie at the Bells Laboratoires in 1972. Despite being an old language, it is still very popular. C is strongly associated with UNIX, as it was developed to write the UNIX operating system. It can be used to develop software like operating systems, databases, compilers etc. It is a structured programming language that is machine-independent and extensively used in various applications, operating systems like Windows and many other complex programs.

FEATURES OF C PROGRAMMING LANGUAGE

1. It is a high-level programming language.

2. It is a structured programming language and it reduces the development time for designing a programming software.

3. It has a rich library: C language has its own library which includes most of the mathematical and logical operations which are predefined. All you have to do it is to include the library you need and then you can execute the functionality of those libraries without having to code them separately.

4. pointers: using pointers, one can directly interact with the physical memory of the computer system

5. Faster Execution of programs: Using C, the users can execute programs faster.

6. Portable: C programs can be executed and run-on different platforms without any or minimal changes.

Advantages of learning C

1. it is one of the most popular programming language in the world

2. Knowing the C language, makes it easy to learn other popular programming languages out there such as C++, C#, Python, Java, JavaScript etc. as the syntax is similar

3. It is a very fast language compared to another like Python, Java, etc.

4. It is a versatile language. It can be used to develop varieties of applications.

**6.) DETAILED STRUCTURE OF THE COMPLETE C PROGRAMMING LANGUAGE**

C language supports various operators, data structures, constructors etc. These features constitute to why C is regarded as a general-purpose programming language because it can be used for system programming, development of compilers, operating systems, games etc.

A C program will usually consist of the following parts:

1. Document Section (Documentation): This consists of the description of the programmer's name, the program and the date it was created. It provides an overview of the program. This is usually written in the form of comments.

2. Preprocessor Section: This contains all the header files used in a program. It prompts C compiler to include the header files before compilation of C programs e.g., "#include <stdio.h>" which contains the definition of stdin, stdout etc. Whenever these definitions are used in the program, the statement #include <stdio.h> needs to be included.

3. Define Section: this section comprises of different constants declared using the define keyword. e.g., #define a = 2

4. Global Declaration: this includes declaration of all the global variable, functions, function declarations in the program which can be accessed anywhere within the program. e.g., float number = 1.02

5. main () function: C program execution starts from the main () function; hence it is MANDATORY to have this function included in every C program.

6. Local Declarations: this refers to variables declared inside a given function or block which can only be accessed within the function or block.

7. Statements: these refers to "if, else, while etc." used while writing program inside the main () function.

8. User Defined Functions: refers to the functions specified by the user for performing a specific task.

9. Return function: this is the last section of a C program. It ends the execution of the main () function.

7.) HOW TO CREATE A C PROGRAMMING FILE ON THE OS

1. Download a text-editor or IDE of your choice e.g., Visual Studio Code, Code Blocks etc.

2. Install the application software after download.

3. Download and install a compiler which will be used to compile the C file. Recommended option is GCC compiler.

4. Load up the text editor or IDE and create a new file

5. Save the new file by giving it a name and ensure the file is saved with the **.c** extension and stored in the location of your choice. Then proceed to write your program.

6. Open up the command prompt on the system and navigate to the directory where the file is saved

7. Type gcc 'filename.c' and press enter to compile the code.

If there are no errors present in the code, an executable file will be generated. Proceed to type out the name of the file in the terminal to execute it and the output of the program will be displayed to the screen.