NAME: Akintayo Lanre Moshood
MATRIC NO: 222460

1. **Write briefly on Unix operating system especially Linux flavor?**

   Unix and Unix-like operating systems are a family of computer operating systems that are derived from the original Unix system from Bell Labs.

   Unix is the most powerful and popular multi-user and multi-tasking Operating System. The basic concepts of Unix originated in the Multics project of 1969. The Multics system was intended as a time-sharing system that would allow multiple users to simultaneously access a mainframe computer.

The Unix operating system is made up of three parts; the kernel, the shell and the programs.

- **The kernel:** The kernel of UNIX is the hub of the operating system: it allocates time and memory to programs and handles the filestore and communications in response to system calls.

- **The shell:** It acts as an interface between the user and the kernel. When a user logs in, the login program checks the username and password, and then starts another program called the shell. The shell is a command line interpreter (CLI).

Unix provides varieties of flavors which is also referred to as Unix-like operating systems that have been developed based on the original UNIX and those flavors differ by fundamental design, commands and features, the hardware platforms (i.e. the processor) for which they are intended and whether they are commercial/proprietary software or free software. Some

Some of the commercial flavor includes;

- **AIX** - developed by IBM for use on its mainframe computers
- **BSD/OS** - a commercial version of BSD developed by Wind River for Intel processors
- **HP-UX** - developed by Hewlett-Packard for its HP 9000 series of business servers
- **IRIX** - developed by SGI for applications that use 3-D visualization and virtual reality
- **QNX** - a real time operating system developed by QNX Software Systems primarily for use in embedded systems
- **Solaris** - developed by Sun Microsystems for the SPARC platform and the most widely used proprietary flavor for web servers
- **Tru64** - developed by Compaq for the Alpha processor

The disadvantages of the commercial/proprietary flavor are that they are very costly, hard to use and they are not free. Hence, it makes it difficult for them them to compete with the Microsoft Windows operating systems except for very heavy-duty corporate applications for which high stability and power were primary considerations.

What saved the Unix-like operating systems as a whole (but not the proprietary UNIXs) was the rapid maturation of a new flavor called Linux. Linux overcame the disadvantages of the proprietary flavors by offering very low cost, the ability to operate on a wide range of platforms (everything from a supercomputer to a wristwatch), the availability of the source code so that it could be easily and freely modified and improved by its users and greater ease of use. This was all accomplished without sacrificing any of the power or stability that characterizes Unix-like operating systems.

Linux has continued to advance rapidly in terms of ease of use and other performance characteristics as a result of its *open source* (i.e., freely available source code) development model, and it has now become the dominant flavor of Unix-like operating systems.

Some of the other free flavor that were made available for everyone includes;

- **FreeBSD** - the most popular of the BSD systems (all of which are direct descendants of BSD UNIX, which was developed at the University of California at Berkeley)
- **NetBSD** - features the ability to run on more than 50 platforms, ranging from acorn26 to x68k
- **OpenBSD** - may have already attained its goal of becoming the most secure of all computer operating systems
- **Darwin** - the new version of BSD that serves as the core for the Mac OS X

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

Software Functional requirements are product features that developers must implement to enable the users to achieve their goals. They define the basic system behavior under specific conditions. In software engineering, functional Requirements are also called **Functional Specification**.

Functional Requirements of a system should include the following things:

- Details of operations conducted in every screen
- Data handling logic should be entered into the system
- It should have descriptions of system reports or other outputs
- Complete information about the workflows performed by the system
- It should clearly define who will be allowed to create/modify/delete the data in the system
- How the system will fulfill applicable regulatory and compliance needs should be captured in the functional document

Some benefits of Functional Requirement include;

- Helps you to check whether the application is providing all the functionalities that were mentioned in the functional requirement of that application

- A functional requirement document helps you to define the functionality of a system or one of its subsystems.
- Errors caught in the Functional requirement gathering stage are the cheapest to fix.
- Support user goals, tasks, or activities

There are different types of functional requirements. They include;

- Transaction Handling
- Business Rules
- Certification Requirements
- Reporting Requirements
- Administrative functions
- Authorization levels
- Audit Tracking
- External Interfaces
- Historical Data management
- Legal and Regulatory Requirements

Example of functional requirements

This is a functional requirement of a bank management system;

For a bank management system, there are two types of modules; the admin module and the user module.

Functional requirements of the admin module;

- **Admin login:** If you login as an Admin then you will be redirected to the Admin Home
- **Add/delete/update account**: The admin can add, delete, update any account.
- **Withdrawal/deposit/statements transaction**: The admin can withdraw, deposit and statement any transaction.
- **Account Information**: The admin can see any account information
- **User details list**: The admin can see user details list
- **Activate/Deactivate account:** The admin can activate and deactivate any account
- **View transaction histories**: The admin can see transaction histories.

Functional requirements for the user module;

- User login, use PIN system
- Creating/open new account registration
- Funds transfer (local/international/domestic)
- View statements transaction

- User account details
- Change Password and PIN

3. **Why does unix often prefer at some points?**

- **Linux is Secure and Private**
  Linux is more secure in comparison to other competing operating systems. Linux requires authorization from the root, or superuser, in the form of a password. Unless the password is known and used, nothing will run – not even a virus! This provides an added benefit of eliminating the need for anti-virus software.

- **Linux is free to use and update**
  Linux is not licensed and is free to download and use. Typical costs for Windows OS range from $55 to $450 dollars depending on the application. There are paid versions of Linux, but these are typically much more affordable.

- **Flexibility for the End-user or Corporate Engineer**

  Linux benefits the advanced engineer just as much as it does the novice user thanks to the ability to set Linux up 'just the way you want it.'
  Whether you're constrained by the hardware you're running on (e.g., it may be older) or you simply want an easy-to-use free operating system; Linux is the solution.

- Linux is easy to install
- Linux is reliable

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

Linux provides a lot of distribution that are designed specifically for science to make scientific operations easy and very efficient. Some of the distributions include;

- **CAELinux 2020:** It is a Linux distribution specifically designed for scientists and IT professionals. It provides all the tools needed in a research setting. It's built on top of the **Glade** toolkit, which makes it easy to use on any Linux system with at least 1 GB of RAM. As a **LiveDVD Linux** distribution, you can boot directly from a DVD or USB flash drive without installation.

- **Fedora Robotic Suite:** It is a complete and specialized software for the hobbyist part of robotics. It provides all the necessary tools for electronics enthusiasts. As a platform-agnostic operating system/robotic development toolkit consisting of several different interfaces and systems, **Fedora Robotic Suite** rapidly simplifies the process of getting started with real robots. Use robots as communication devices, and develop robotic applications.

- **Lin4Neuro:** It is a scientific Linux distribution is a niche contender on this list with a history dating back several years and origins in Japan. As a Linux distribution designated for use in the field of neuroimaging analysis. **Lin4Neuro** is categorically regarded as the de facto operating system for specialists in the field of neuroscience as the operating system is by a seasoned veteran in the field who is apparently available to support users of the same caliber.

5. **What type of programming language is C?**
   C is a structured, procedural programming language that has been widely used both for operating systems and applications and that has had a wide following in the academic community.

   C is called structured programming language because a program in C language can be divided into small logical functional modules or structures with the help of function procedure.

   It is called a procedural programming language because a program in C follows a **procedure** of steps written in it, **called** functions. It follows a top-down approach i.e. Much importance is given to flow of program rather than on data on which functions operate.

6. **Give the detailed structure of a complete C programming language**
   The structure is divided into 6 parts;

   - **Documentation**

     This section consists of the description of the program, the name of the program, and the creation date and time of the program. It is specified at the start of the program in the form of comments. Documentation can be represented as:

     ```
     /*
         description, name of the program, programmer name, date, e.t.c
     */
     Or using single line comment like;
     // description, name of the program, programmer name, date, e.t.c
     ```

   - **Preprocessor Section**

     All the header files of the program will be declared in the preprocessor section of the program. Header files help us to access other's improved code into our code.

     Example;

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

   - **Definition**

     Preprocessors are the programs that process our source code before the process of compilation. Preprocessor directives start with the '#' symbol. The #define preprocessor is used to create a constant throughout the program. Whenever this name is encountered by the compiler, it is replaced by the actual piece of defined code.

Example;

```
#define X 20
```

- **Global Declaration**

  The global declaration section contains global variables, function declaration, and static variables. Variables and functions which are declared in this scope can be used anywhere in the program.

  Example;

  ```
  int num = 18;

  int sum(int y);
  ```

- **Main() Function**

  Every C program must have a main function. The main() function of the program is written in this section. Operations like declaration and execution are performed inside the curly braces of the main program.

  Example;

  ```
  void main();
  ```

- **Sub Programs**

  User-defined functions are called in this section of the program.

  Example;

  ```
  int sum(int x, int y)
  {
      return x+y;
  }
  ```

7. **How can I create a C programming file on the OS**
   1. Go to a supported code editor. Visual studio code is highly recommended.
   2. In the menu bar, click on files and create a new file. Make sure the file ends with a c extension.
   3. Install all necessary extensions required to make your code run. Some of them includes; code runner, c/c++ compiler
   4. Make sure you have gcc compiler installed on your OS. gcc compiler is used to compile and execute your code.