**Name: Tamilore Muyideen ADEBAYO**

**Matric No: 230868**

**Level: 200**

**Course CSC235 Object Oriented Programming**

**1.0 Unix Operating System**

Unix is a stable, multi-user, multi-tasking Operating System (OS) used on servers, desktops, laptops, mobile devices and other embedded systems. It was developed in the late 1960s by Bell Labs (now AT&T) computer scientists, Ken Thompson and Dennis Ritchie, continuing a project originally started by Bell Labs, General Electric and the Massachusetts Institute of Technology. Together with some other Bell Labs researchers, the scientists-duo also built some components to work with the OS such as a command-line interface, a hierarchical file system and some other small application softwares.

Before 1973, Unix was written in assembly language however in 1973, the OS’s fourth edition was written in the C programming language. This was phenomenal at the time as it was thought that operating systems were too complex to be written in a high-level programming language. In 1991, a university of Helsinki student, Linus Torvalds, created Linux for his PC based on UNIX.

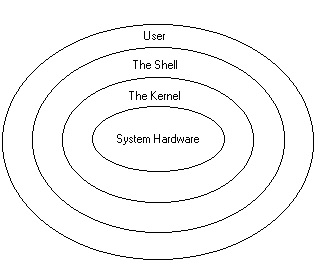
**1.1 Types of Unix**

Unix comes in different types. Some are proprietry and require permission or a license key has to be obtained before they can be used such as the IBM Advanced Interactive eXecutive (AIX) or Oracle Solaris while others are free and open source such as OpenDSB and FreeDSB

**1.2 The Unix Shell and Kernel**

The Unix kernel is the core of the operating system i.e the most important part. It interacts with hardware components, takes care of memory, processes and I/O management. The kernel is also responsible for handling interrupts and errors, maintaining the file system and allocating resources to the users

The Unix shell serves as a command-line intermediary between the user and the kernel. It is used by the OS for system execution based on user input via ran commands, programs and shell scripts. Just as operating systems, there are different flavours of a shell; each flavour having its own group of known commands and functions. The **$** is known as the shell prompt. While it is displayed, a user can type in commands. After typing in the command and pressing the enter key, the shell determines the command to be executed by examining the first word.



**1.3 The Linux Operating System**

The Linux operating system is the most popular type of Unix operating system. It is open source and free to use. The os is based on the linux kernel developed by Linus Torvalds and first released in September 1991. A few of the basic features of the operating system include:

* **Portability:** The Linux operating system can be installed and used on any kind of hardware
* **Multiuser:** The os supports use by more than one user who can all have access to memory and application softwares
* **Security:** The Linux os has great security features such as data encryption and password protection

**1.4 Linux Distributions**

A Linux distribution (distro) is an operating system developed from a group of software which consists of the Linux Kernel, a window system, a package management system, GNU tools and additional libraries and softwares. Users usually download any of the distributions to get their operating system.

Linux os comes in various distributions; popular among them are Fedora linux, Debian and Ubuntu which, also has many distributions and modifications such as Linux Mint, Lubuntu and Xubuntu. Commercial distributions include SUSE Linux and Red Hat Linux. There are also things such as a desktop environment (such as KDE Plasma, GNOME) and a windowing system (such as Wayland, X11). Server based distributions may exclude graphics completely.

**2.0 Software Functional Requirements**

Requirement analysis is a very inportant aspect of software development. It enables he success of a software project to be tested. Requirements can be categorized into functional and non-functional requirements.

Functional requirement is a basic description of the various functions that must be offered by a software or its components. This includes various inputs to the system, system behaviour and processes and the outputs gotten from the system. Communication of functional requirements is done using statements ranging from high-level abstract statements to concise mathematical statements. Details contained in a software functional requirement include:

* Details of system workflow
* Operations carried out on every screen
* Authentication details for CRUD data processes
* Detailed system reports
* Concise documentation of the system

**2.1 Types of Functional Requirements**

There are different types of functional requirements based on the aspect of te software that is being described. Some of them are:

* Data management requirements
* Authorization requirements
* Administrative requirements
* Legal and regulatory requirements
* Transaction handling
* External interface requirements

A great benefit of software functional requirement is that missing requirements are easily noticed and errors detected in this stage of software development are the easiest anc cheapest to fix.

Some examples of software functional requirements are:

* The software should be integrated with finance APIs
* Only managerial level staff have access to user details
* All white colours used in the system must have #eeeeee as color code

3. Unix is preferred for the following reasons

* It is open source therefore the code can be easily accessed and modified
* The operating system and other application software are free to use.
* Unix can be used on a variety of devices such as routers, IoT systems, web servers, etc.
* Unix is free from viruses
* Unix has greater built-in security and permission features
* Unix os can run for extremely long periods without needing to restart
* The Unix os is backed by a large community for support

4. Unix is referred to as a scientist operating system because unlike some other os, there is a quite steep technical curve to learning how to use the Unix os. The os also handles things like file wrangling, big data manipulation, web hosting, process parallelization and automation better than other os whilst utilizing less system resources.

5. C is a high-level and general purpose programming language. The language is an imperative procedural language which supports structured programming, lexical variable scope and recursion. The language has a static type system. It is a compiled language.

6. Structure of a C program

The structure of a program guides how the program should be written. Any program written without this structure will experience compilation errors. The structure of a C program is as follows:

Header: #include <stdio.h>

#include <string.h>

Main: int main() {

variable int x =2, y = 10;

declaration: int z = x + y;

body: printf(“the sum of x and y is %d”, z);

printf(“the end”);

Return: return 0;

}

**Header file:** this contains C functions and macros to be shared among several source files. Examples are: stddef.h, stdlib.h, math.h, etc.

**The main method:** This is where the C compiler starts the execution of the source code. This contains the body of the code and other function calls. Its syntax is

int main() {}

**Variable declaration:** this is the section that all variables to be used in the program are declared and may be initialized. This does not mean variables can not be declared and initialized at other parts of the program. Note that no variable can be used without its declaration.

**Body:** this is where operations to be carried out by the function is written. It can be printing, conditions, sorting, searching, etc. Each statement is terminated by a semicolon (;).

**Return statement:** this refers to the value that is given by the function after its execution. The return statement and value depends on the return type of the function.

7. Steps to creating a C program assuming that you have the compiler (gcc) installed.

* Open the terminal by searching in the apps drawer or using keyboard shortcut Ctrl + Alt + t
* Navigate to the folder where you want to create the C file using the cd command

e.g: cd documents/cfolder

* Use the touch command to create the file

e.g: touch hello.c (where hello is the name of the file)

* Open the file with your preferred editor or IDE

e.g: code hello.c