Finolex Academy of Management and Technology, Ratnagiri			
Department of Information Technology			
Subject name: **UNIX Lab**			Subject Code: **ITL402**
Class	SE IT	Semester – IV (CBCGS)	Academic year: 2022-2023
Name of Student			**QUIZ Score :**
Roll No		Assignment/ExperimentNo.	**01**
Title: To study history and architecture of UNIX operating system			

1. Course objectives applicable: LO1

2. Course outcomes applicable:PO1 , PEO1

3. **Learning Objectives:**
To learn history and architecture of UNIX operating system

4. Practical applications of the assignment/experiment:
Understanding architecture of UNIX operating system

5. Prerequisites:
C Programming Language and Operating System

6. Hardware Requirements:
PC with minimum 2GB RAM

7. Software Requirements:
Fedora installed.

8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/Marks obtained):
When was UNIX evolved?
What is architecture of unix?

9. Experiment/Assignment Evaluation:

Sr. No.	**Parameters**	**Marks obtained**	**Outof**
1	Technical Understanding (Assessment may be done based on Q &A **or** any other relevant method.) Teacher should mention theother method used -		6
2	Neatness/presentation		2
3	Punctuality		2

Date of performance (DOP)		**Total marks obtained**		**10**
Date of checking (DOC)		**Signature of teacher**		

HISTORY OF THE UNIX OPERATING SYSTEM

The first version of UNIX was created in 1969 by Kenneth Thompson and Dennis Ritchie, system engineers at AT&T's Bell Labs. It went through many revisions and gained inpopularity until 1977, when it was first made commercially available by Interactive SystemsCorporation.

At the same time a team from the University of California at Berkeley was working toimprove UNIX. In 1977 it released the first Berkeley Software Distribution, which becameknown as BSD. Over time this won favour through innovations such as the C shell.

Meanwhile the AT&T version was developing in different ways. The 1978 release of Version 7 included the Bourne Shell for the first time. By 1983 commercial interest was growing and Sun Microsystems produced a UNIX workstation. System V appeared, directlydescended from the original AT&T UNIX and the prototype of the more widely used variant today

MODERN VARIANTS OF UNIX

There are two main versions of UNIX in use today: System V and BSD. System V is themore popular of the two.

From a user's perspective they are very similar and you are unlikely to have difficultyunless you use more than one type of system. In this case you might notice differences inthe structure of the file system or in how certain commands behave. The on-line manual should be helpful if you have problems.

Although UNIX help is now based on a System V variant of UNIX, you should see onlyminor differences in the example input and output if your system is a BSD one.[Home] [Search] [Index]

CONNECTION BETWEEN C AND UNIX

If you think C programming and Unix are unrelated, then you are making a big mistake. Back in the 1970s and 1980s, if the Unix engineers at Bell Labs had decided to useanother programming language instead of C to develop a new version of Unix, then wewould be talking about that language today.

The relationship between the two is simple; Unix is the first operating system that isimplemented with a high-level C programming language, got its fame and power from Unix. Of course, our statement about C being a high-level programming language is not true intoday's world.

This article is an excerpt from the book Extreme C by Kamran Amini. Kamran teachesyou to use C's power. Apply object-oriented design principles to your procedural C code.
You will gain new insight into algorithm design, functions, and structures. You'll alsounderstand how C works with UNIX, how to implement OO principles in C, and what multiprocessing is.

I do not think we can find anyone better than Dennis Ritchie himself to explain why Cwas invented after the difficulties met with B. In this section, we're going to list the causesthat prompted Dennis Ritchie, Ken Thompson, and others create a new programming language instead of using B for writing Unix.Limitations of the B programming language: Bcould only work with words in memory: Every single operation should have been performedin terms of words. Back then, having a programming language that was able to work withbytes was a dream. This was because of the available hardware at the time, which addressed the memory in a word-based scheme.

The difficulties with B, particularly its slow development and execution on machinesthat were available at the time, forced Dennis Ritchie to develop a new language. This newlanguage was called

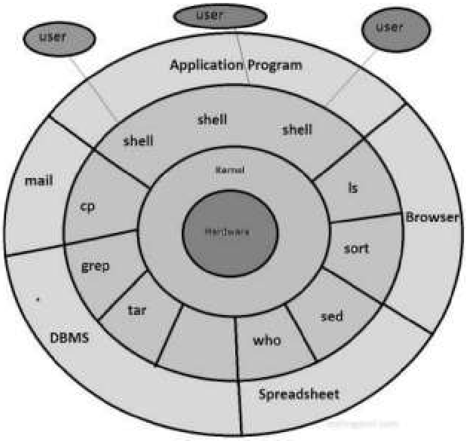
NB, or New B at first, but it eventually turned out to be C.

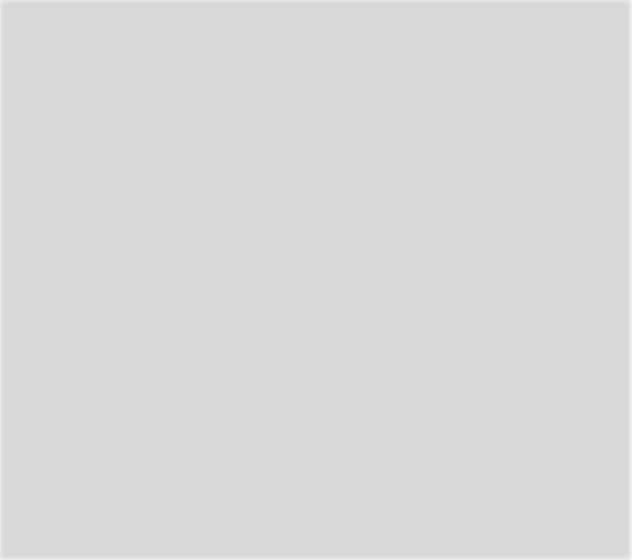
This newly developed language, C, tried to cover the difficulties and flaws of B andbecame a de facto programming language for system development, instead of the assemblylanguage. In less than 10 years, newer versions of Unix were completely written in C, and all newer operating systems that were based on Unix got tied with C and its crucial presence inthe system.

As you can see, C was not born as an ordinary programming language, but instead, itwas designed by having a complete set of requirements in mind. You may consider languages such as Java, Python, and Ruby to be higher-level languages, but they cannot beconsidered as direct competitors as they are different and serve different purposes. For instance, you cannot write a device driver or a kernel module with Java or Python, and theythemselves have been built on top of a layer written in C.

Unlike some programming languages, C is standardized by ISO, and if it is required tohave a certain feature in the future, then the standard can be modified to support the newfeature.

BASIC ARCHITECTUE OF UNIX OPERATINGSYSTEM





UNIX operating system is like a layered one we first find H/W block next the heart of anoperating system i.e. KERNEL, then SHELL, and then users hence with this architecture thereis something new called shell which we don't find in any other operating system because of which there won't be any bugs in the system the detail structure can be thought in the classwith pictorial presentation.

HARDWARE: - Hardware is nothing but the system components which can be seen withhuman eye or can be touched like monitor, keyboard, hard disk etc..,

KERNEL: - The kernel is the heart of the operating system. It interacts with the hardware andmost of the tasks like memory management, task scheduling and file management.

SHELL :- The shell is the utility that processes your requests. When you type in a commandat your terminal, the shell interprets the command and calls the program that you want. Theshell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are themost famous shells which are available with most of the Unix variants. Shell maintains twodirectories. They are :

/sbin : It contains all super user executable commands.

/bin : It contains all normal user executable commands

12. Learning Outcomes Achieved:

Students will be able to study history and architecture of UNIX operating system

13. Conclusion:

Unix has deep history which helps understand its effective functionality and architecture

14. References :

[1] Unix, concepts and applications by Sumitabha Das, McGraw-Hill
[2] Mastering Shell Scripting, Randal. K. Michael, Second Edition, Wiley Publication