Overview of Unix OS

Varun Saluja, Vanita
Dronacharya College of Engineering

ABSTRACT: **UNIX is an operating system developed by AT&T's Bell Laboratories. It's a very mature operating system known for its advantages such as stability and reliability. An early multi-user and multi-tasking operating system which grows up alongside various networking technologies and makes it an early mainstay of Internet networking and communication. UNIX is stable because of its ARCHITECTURE. It is designed based on hardware and software layers. At the base of its architecture is the hardware on which the software runs. A kernel layer sits between higher computer functions and the hardware. Applications and commands interact with the kernel. UNIX is used as a SERVER OS. As a server, its function is to allow multiple users to access different data and functions on the host computer. UNIX WORKSTATIONS are used in engineering, computer-aided design and modeling, animation and graphic design. Many of these workstations use one of many different windowing systems or GUIs. This review paper covers a basic study about UNIX OPERATING SYSTEM, its architecture, servers, workstations and its functions.**

INDEXED TERMS: *Architecture, Server OS, Workstations, Kernel, Shell & GUI.*

I. INTRODUCTION

UNIX is an operating system which was first developed in the 1960s, and has been under constant development ever since. By operating system, we mean the suite of programs which make the computer work. It is a stable, multi-user, multi-tasking system for servers, desktops and laptops.

UNIX systems also have a graphical user interface (GUI) similar to Microsoft Windows which provides an easy to use environment. However, knowledge of UNIX is required for operations which aren't covered by a graphical program, or for when there is no windows interface available, for example, in a telnet session. Created by just a handful of programmers, UNIX was designed to be a small, flexible system used exclusively by programmers.

UNIX was one of the first operating systems to be written in a high-level programming language, namely C. This meant that it could be installed on virtually any computer for which a C compiler existed. This natural portability combined with its low price made it a popular choice among universities. (It was inexpensive because antitrust regulations prohibited Bell Labs from marketing it as a full-scale product.)

Bell Labs distributed the operating system in its source language form, so anyone who obtained a copy could modify and customize it for his own purposes. By the end of the 1970s, dozens of different versions of UNIX were running at various sites.

II. TYPES OF UNIX OS

Unix is not a single operating system. It has many flavors (aka. variants, types, or implementations). Although based on a core set of Unix commands, different flavors have their own unique commands and features, and designed to work with different types of hardware. No one knows exactly how many Unix flavors are there, but it is safe to say that if including all those that are obscure and obsolete, the number of Unix flavors is at least in the hundreds. You can often tell that an operating system is in the Unix family if it has a name that is a combination of the letters U, I, and X.

The following is some of the well-known Unix flavors, with links to their official home pages.

1. AIX by IBM
2. BSD/OS (BSDi) by Wind River
3. CLIX by Intergraph Corp.
4. Debian GNU/Linux by Software in the Public Interest, Inc.
5. Tru64 Unix (formerly Digital Unix) by Compaq Computer Corp.

6. DYNIX/ptx by IBM (formerly by Sequent Computer Systems)
7. Esix Unix Esix Systems
8. FreeBSD by FreeBSD Group
9. GNU Herd by GNU Organization
10. HAL SPARC64/OS by HAL Computer Systems, Inc.
11. HP-UX by Hewlett-Packard Company
12. Irix by Silicon Graphics, Inc.
13. Linux by several groups several
14. LynxOS by Lynx Real-Time Systems, Inc.
15. MacOS X Server by Apple Computer, Inc.
16. NetBSD by NetBSD Group
17. NonStop-UX by Compaq Computer Corporation
18. OpenBSD by OpenBSD Group
19. OpenLinux by Caldera Systems, Inc.
20. Openstep by Apple Computer, Inc.
21. Red Hat Linux by Red Hat Software, Inc.
22. Reliant Unix by Siemens AG
23. SCO Unix by The Santa Cruz Operation Inc.
24. Solaris by Sun Microsystems
25. SuSE by S.u.S.E., Inc.
26. UNICOS by Silicon Graphics, Inc.
27. UTS by UTS Global, LLC

III. THE UNIX OPERATING SYSTEM

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

The Kernel:

In computing, the **kernel** is a computer program that manages input/output requests from software, and translates them into data processing instructions for the central processing unit and other electronic components of a computer. The kernel is a fundamental part of a modern computer's operating system.

When a computer program (in this context called a process) makes requests of the kernel, the request is called a system call. Various kernel designs differ in how they manage system calls and resources. For example, a monolithic kernel executes all the operating system instructions in the same address space in order to improve the performance of the system. A microkernel runs most of the operating system's background processes in user space, to make the operating system more modular and, therefore, easier to maintain.[3]

Because of its critical nature, the kernel code is usually loaded into a protected area of memory, which prevents it from being overwritten by other, less frequently used parts of the operating system or by application programs. The kernel performs its tasks, such as executing processes and handling interrupts, in kernel space, whereas everything a user normally does, such as writing text in a text editor or running programs in a GUI (graphical user interface), is done in user space. This separation is made in order to prevent user data and kernel data from interfering with each other and thereby diminishing performance or causing the system to become unstable (and possibly crashing).

Shell:

A Unix shell is a command-line interpreter or shell that provides a traditional user interface for the Unix operating system and for Unix-like systems. Users direct the operation of the computer by entering commands as text for a command line interpreterto execute, or by creating text scripts of one or more such commands. Users typically interact with a Unix shell using a terminal emulator, however, direct operation via serial hardware connections, or networking session, are common for server systems.

The most influential Unix shells have been the Bourne shell and the C shell. These shells have both been used as the coding base and model for many derivative and work-alike shells with extended feature sets.

The Bourne shell, sh, was written by Stephen Bourne at AT&T as the original Unix command line interpreter; it introduced the basic features common to all the Unix shells, including piping, here documents, command substitution, variables, control structures for condition-testing and looping and filename wildcarding. The language, including the use of a reversed keyword to mark the end of a block, was influenced by ALGOL 68.

The C shell, csh, was written by Bill Joy while a graduate student at University of California,

Berkeley. The language, including the control structures and the expression grammar, was modeled on C. The C shell also introduced a large number of features for interactive work, including the history and editing mechanisms, aliases, directory stacks, tilde notation, cdpath, job control and path hashing.

IV. PROGRAMS

A computer program, or just a program, is a sequence of instructions, written to perform a specified task with a computer. A computer requires programs to function, typically executing the program's instructions in a central processor. The program has an executable form that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled), enables a programmer to study and develop its algorithms. A collection of computer programs and related data is referred to as the software.

Computer source code is typically written by computer programmers.[3] Source code is written in a programming language that usually follows one of two main paradigms: imperative or declarative programming. Source code may be converted into an executable file(sometimes called an executable program or a binary) by a compiler and later executed by a central processing unit. Alternatively, computer programs may be executed with the aid of an interpreter, or may be embedded directly into hardware.

Computer programs may be ranked along functional lines: system software and application software. Two or more computer programs may run simultaneously on one computer from the perspective of the user, this process being known as multitasking.

V. THE DIRECTORIES (. AND ..)

The current directory (.)

In UNIX, (**.**) means the current directory, so typing

%cd .

means stay where you are.

This may not seem very useful at first, but using (**.**) as the name of the current directory will save a lot of typing, as we shall see later in the tutorial.

The parent directory (..)

(**..**) means the parent of the current directory, so typing

%cd ..

will take you one directory up the hierarchy (back to your home directory).

Note: typing **cd** with no argument always returns you to your home directory. This is very useful if you are lost in the file system.

VI. PATHNAMES

pwd (print working directory)

Pathnames enable you to work out where you are in relation to the whole file-system. For example, to find out the absolute pathname of your home-directory, type **cd** to get back to your home-directory and then type

%pwd

The full pathname will look something like this – **/home/its/ug1/ee51vn**

which means that ee51vn (your home directory) is in the sub-directory ug1 (the group directory),which in turn is located in the its sub-directory, which is in the home sub-directory, which is in the top-level root directory called " / ".

VII. UNIX COMMANDS

Files and Directories:

These commands allow you to create directories and handle files.

Command	Description
cat	Display File Contents
cd	Changes Directory to dirname
chgrp	change file group
chmod	Changing Permissions
cp	Copy source file into destination
file	Determine file type
find	Find files
grep	Search files for regular expressions.
head	Display first few lines of a file
ln	Create softlink on oldname
ls	Display information about file type.
mkdir	Create a new directory dirname
more	Display data in paginated form.
mv	Move (Rename) a oldname to newname.
pwd	Print current working directory.
rm	Remove (Delete) filename
rmdir	Delete an existing directory provided it is empty.
tail	Prints last few lines in a file.
touch	Update access and modification time of a file.

Misc Commands:

These commands list or alter information about the system:

Command	Description
chfn	Change your finger information
chgrp	Change the group ownership of a file
chown	Change owner
date	Print the date
determin	Automatically find terminal type
du	Print amount of disk usage
echo	Echo arguments to the standard options
exit	Quit the system
finger	Print information about logged-in users
groupadd	Create a user group
groups	Show group memberships
homequota	Show quota and file usage
iostat	Report I/O statistics
kill	Send a signal to a process
last	Show last logins of users
logout	log off UNIX
lun	List user names or login ID
netstat	Show network status
passwd	Change user password
passwd	Change your login password
printenv	Display value of a shell variable
ps	Display the status of current processes
ps	Print process status statistics
quota -v	Display disk usage and limits
reset	Reset terminal mode
script	Keep script of terminal session
script	Save the output of a command or

	process
setenv	Set environment variables
stty	Set terminal options
time	Time a command
top	Display all system processes
tset	Set terminal mode
tty	Print current terminal name
umask	Show the permissions that are given to view files by default
uname	Display name of the current system
uptime	Get the system up time
useradd	Create a user account
users	Print names of logged in users
vmstat	Report virtual memory statistics
w	Show what logged in users are doing
who	List logged in users

VIII. SUMMARY

- UNIX is a computer operating system.
- An operating system is the program that controls all the other parts of a computer system, both the hardware and the software. It allocates the computer's resources and schedules tasks. It allows you to make use of the facilities provided by the system. Every computer requires an operating system.
- UNIX is a multi-user, multi-tasking operating system. Multiple users may have multiple tasks running simultaneously. This is very different from PC operating systems such as MS-DOS or MS-Windows (which allows multiple tasks to be carried out simultaneously but not multiple users).
- UNIX is a machine independent operating system. Not specific to just one type of computer hardware. Designed from the beginning to be independent of the computer hardware.
- UNIX is a software development environment. Was born in and designed to function within this type of environment.
- The "UNIX" trademark, previously owned by AT&T and then deeded to UNIX Systems Laboratories (USL), an AT&T subsidiary, passed to Novell when it acquired USL. After a brief period of negotiations with rival Unix vendors, namely, Sun Microsystems, Santa Cruz Operation, International Business Machines, and Hewlett-Packard, Novell granted exclusive licensing rights of the UNIX trademark to X/Open Co. Ltd., an Open Systems industry standards branding agent based in the United Kingdom.