

History of UNIX

1

The Unix Operating System

- Unix is a multi-user, multi-tasking, layered O/S
 - Innermost layer is the hardware that provides the services for the O/S.
 - The O/S (kernel) interacts directly with the hardware and provides services to upper layers
 - Programs interact with the kernel through a set of standard system calls.
 - Services: accessing a file: open, close, read, write, link, or execute a file; changing ownership of a file or directory; creating, suspending, or killing a process; enabling access to hardware devices; setting limits on system resources.

4

1965 – Bell Labs joins w/MIT and GE in the development of Multics (Multi-user, multi-processor, and multi-level -hierarchical - file system) O/S.

1969 – AT&T drops out Multics Project
Ken Thompson, Dennis Ritchie, et al. designed and implemented first version of Unix File System on a PDP-7.

1970, Jan 1 – time zero for Unix

1971 – Used as a text processing tool for Bell Labs patent department

1973 – Rewritten mostly in C, a new language developed by Dennis Ritchie.

1974 – Dennis and Ritchie publish a paper in the 'Communications of the ACM' describing Unix₂ O/S.

Unix File System

- It looks like an inverted tree structure.
- Each node is either a file or a directory of files.
- You specify a file or directory by its path name
 - full/absolute path starts with the root, /.
 - relative path starts at current working directory.
- Two special directory entries:
 - . The current directory
 - .. The parent of the current directory

5

1977 – 500 Unix sites world-wide.

1980 – BSD 4.1 (Berkeley Software Development)

1983 – SunOS, BSD 4.2, Sys V.

1984 – 100,000 Unix sites world-wide

1988 – AT&T and Sun develop System V Release 4 (SVR4)

1993 – Novell buys Unix from AT&T

1995 – Santa Cruz operations buys UnixWare from Novell – SCO and HP plan to develop 64-bit version of Unix.

1997 – About 3 million Unix systems shipped world-wide.

3

Unix Directories, Files and Inodes

- Every directory and file is listed in its parent directory.
- A directory is a file that contains a table listing the files contained within it
- An Inode is a special file designed to be read by the kernel to get details of a file:
 - Permissions, ownership, date of creation, last access and change, physical locations of the data blocks on the disk containing the file.
- The system does not require any particular structure for the data in the file itself.
 - No header, trailer, label information, or EOF char.

6

Unix Programs

- A program or command interacts with the kernel to provide the environment and perform the functions called for by the user:
 - Executable shell file (shell script)
 - Built-in shell command
 - A source compiled, object code file.

The shell is a command interpreter. The user interacts with the kernel through the shell.

- Shell programs are binary (usually compiled from C source code) and placed in /bin.

7

Identity

- The system identifies you by the user and group numbers (userid and groupid, respectively)
 - Your primary group is the one associated with your username in the password database file
 - You may belong to more than one group

```
% id
```

```
% groups
```

10

Logging In and Out

A user is prompted for **login** username and then a **password**.

- Unix is case sensitive!
- ^D – indicates end of data stream
- ^C – interrupt
- logout – leave the system
- exit – leave the shell

8

Unix Command Line Structure

- A command is a program that tells the Unix system to do something
command [options] [arguments]
 - option changes the way command performs
 - argument indicates on what the command is to perform its action

```
% ls -alR
```

```
% ls -a -l -R
```

```
% lpr -Pprinter3 -#2 file
```

11

Terminal Type

- Most systems are set up so that the user is prompted for a terminal type
 - vt100, sun, xterms, xterm
 - Terminal type indicates to the Unix system how to interact with the session just opened

```
% setenv TERM <term type> (C shell)
```

9

Control Keys

- Control Keys are used to perform special functions on the command line or within an editor.
- ^Q and ^S : stop and start signal
- ^U : line kill signal

12

Getting Help

- The Unix manual, usually called man pages, is available on-line to explain the usage of the Unix system and commands.

man [options] command_name

-k keyword

-M path

-a

13

File Permissions

%ls -lg

-rwxr-x--- user unixgroup size Month nn hh:mm filename

chmod nnn [argument list] numeric mode

chmod [who]op[perm] [argument list] symbolic mode

-R recursively descend through the directory structure and change the modes

% chmod 755 file1 or % chmod u=rwx, go=r-x file1

16

Directory Navigation and Control

- cd [directory]
- ls [options] [directory or file]
- mkdir [options] [directory]
- pwd
- rmdir [options] directory

14

Variables

- A number of variables are automatically set by the shell when it starts. These allow you to reference arguments on the command line.

\$#

\$-

\$?

\$\$

process number of the current process

\$!

\$n argument on the command line, 1 <= n <= 9

\$0 name of current shell or program

\$* all arguments on the command line "\$1 \$2 ... \$9"

@\$@ all arguments on the command line "\$1" "\$2" ... "\$9"

\$argv[n]

\$#argv report the number of words in the input list

17

File Maintenance Commands

- chgrp [options] group file
- chmod [options] file
- chown [options] owner file
- cp [options] file1 file2
- mv [options] file1 file2
- rm [options] file

15