

The Unix Way

Prasad

Contents

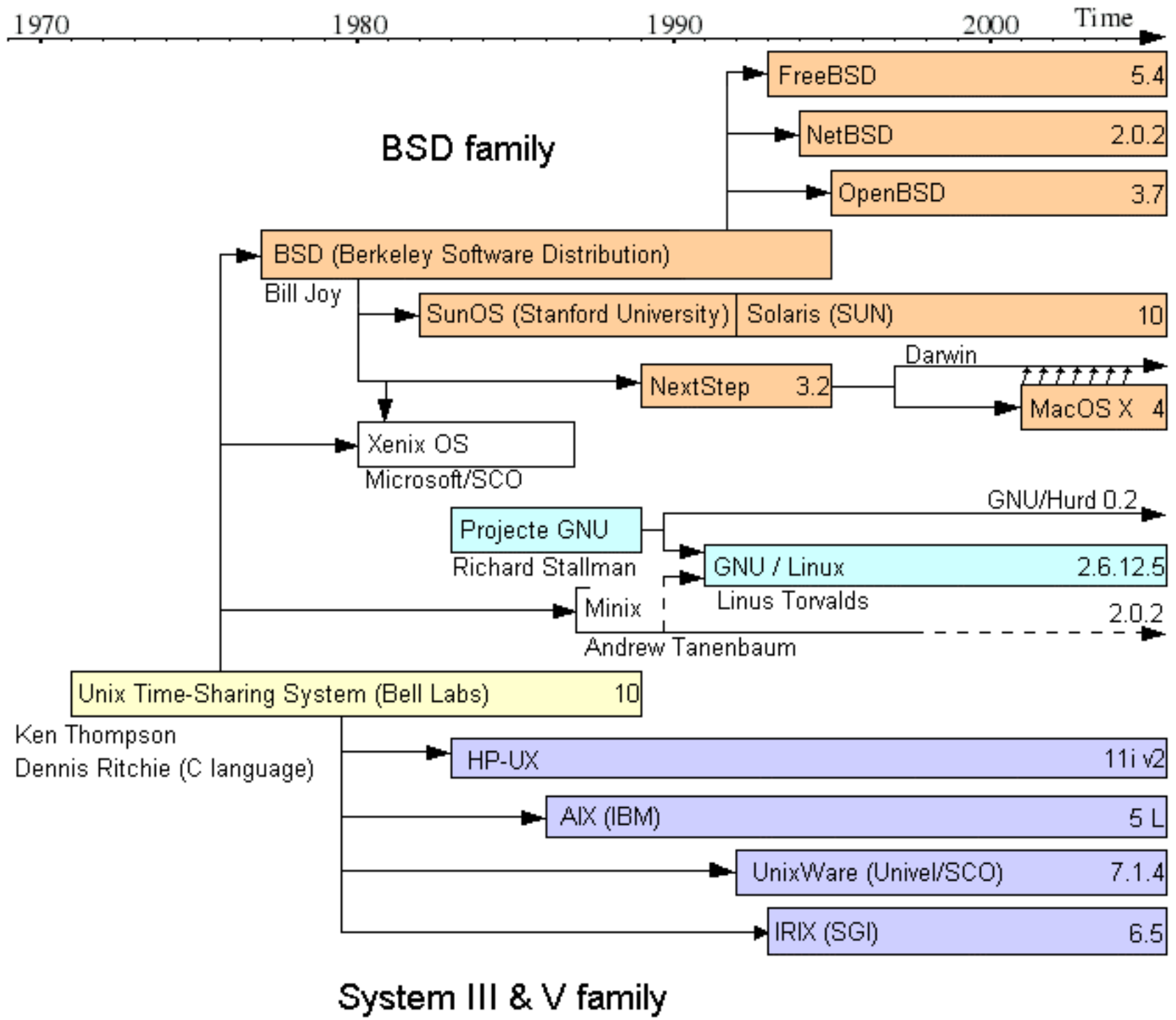
- Origins
- Flavours and History
- A Unix System
- Unix Shell and Commands
- Combining Commands
- **Brief Introduction** to managing your GNU/Linux

Origins

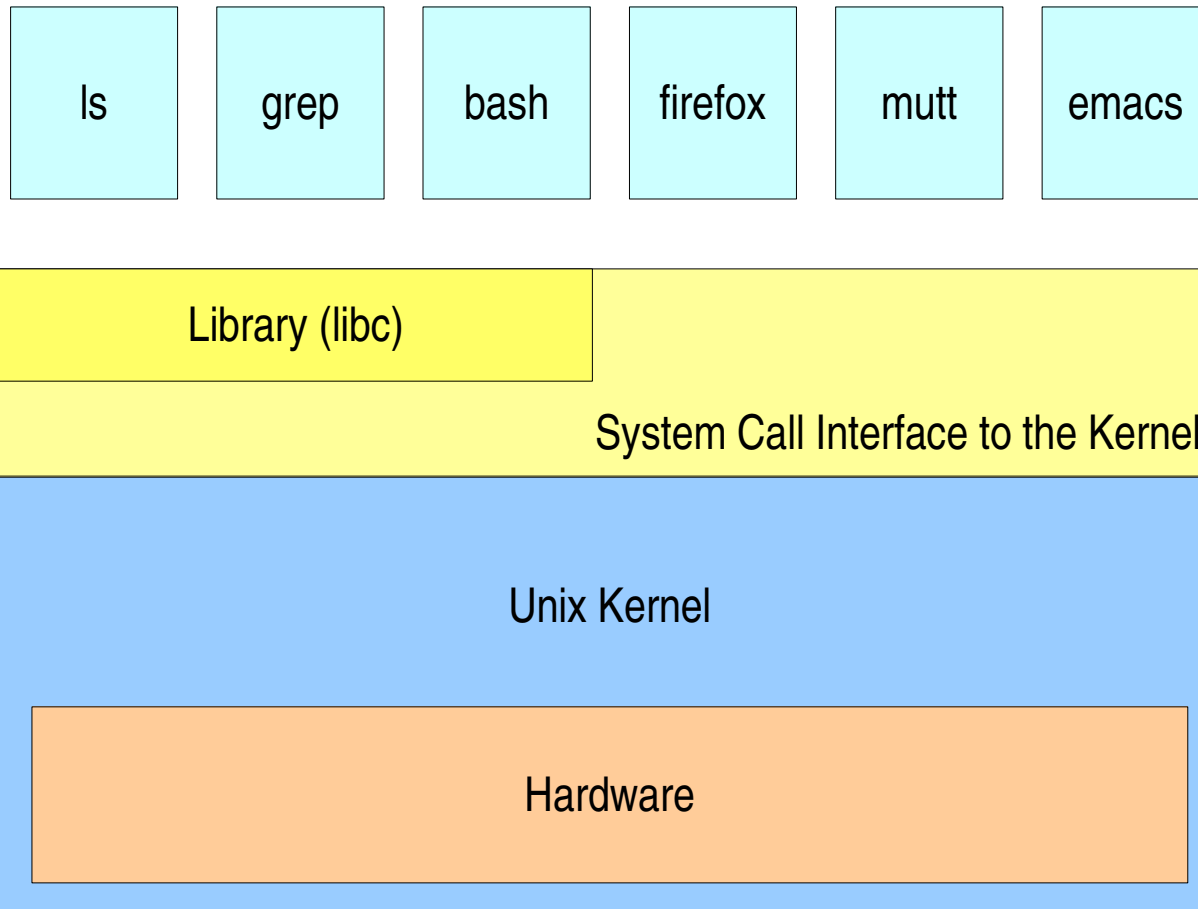
- AT&T Bell Laboratories
- Ken Thompson
- Dennis Ritchie
- Late sixties and the early seventies
- AT&T Bell continued development till the nineties

Flavours and History

- What is Unix in the present day?
- Unix is now better known by the other 'unix-like' operating systems
- Most of the famous operating systems are unix-like
- GNU/Linux, Solaris, FreeBSD, NetBSD, OpenBSD, HP/UX, AIX, SCO UnixWare, SGI IRIX... **Unix is now everywhere**



The Unix System



The Unix System

- Hardware
- Kernel lies over the Hardware
- Over the kernel are the libraries
- Every piece of hardware is a file. Permissions to the file dictate permissions to access the hardware!
- The shell, commands and other executables use hardware through the libraries and kernel
- For the user, shell is the interface that takes the commands and gives back the results

Unix Shell and Commands

- Shell is a program that presents the user with an interpreted programming environment
- A Skeleton Shell
 - Show prompt
 - Take command
 - Execute the command
 - Show prompt ... the cycle goes on
 - **The actual shells have many, seriously many more features**
- Examples: ash, bash, ksh and csh

Unix Shell and Commands

- A Unix shell
 - Has variables and built-in commands
 - Executes external commands (“programs”)
 - Is capable of redirecting input and output from commands
- Unix is a toolbox
 - Each tool has a simple job.
 - Tools can be combined to get a more specialized job; The redirection provided by shells are very useful here

Unix Shell and Commands

- Some shell built-in commands in **bash**
 - `cd`
 - `echo`
 - `eval`
 - `exit`
 - `bg`, `fg`, `jobs`
 - Job control; `bg` sends a process to background, `fg` brings a command in the background to foreground and `jobs` lists all the background processes running in this shell.

Unix Shell and Commands

- Some Unix Commands
 - `ls`: List contents of directory
 - `rm`: Remove files and directories
 - `touch`: Create empty files
 - `mkdir`: Create directories
 - `rmdir`: Remove directories
 - `find`: Find files and directories

Unix Shell and Commands

- Some more Unix Commands/Tools
 - `grep`: looks for a pattern in input stream
 - `cut`: cuts in the input stream based on a separator
 - `paste`: pastes two or more files horizontally
 - `sort`: sorts the input stream
 - `wc`: counts the number of lines and chars in the input stream
 - `sed`: used to edit input stream on a per-line basis
 - `ps`: prints out the list of processes running in the system

Unix Shell and Commands

- Background Processes

- Lets you reuse shell when a command is already running
- '&' operator when given at the end of the command sends it to the background in bash
- Stopping(Pausing) a running process and giving the command 'bg' does the same. Giving 'fg' at the command prompt gets it to the foreground again.
- When multiple processes when sent to background 'fg' takes an integer argument – serial number

Unix Shell and Commands

- Examples

- `grep "prasad" /etc/passwd`

- `cut -f1 -d: /etc/passwd`

- `paste /etc/passwd /etc/shadow`

- `sort /etc/passwd &`

- `wc -l /etc/passwd`

- `sed -e "s/prasad/tux/" /etc/passwd`

- Format of /etc/passwd (**remember this format**)

- `username:password:uid:gid:details:homedir:shell`

Combining Commands

- Why combine commands?
 - Each tool in Unix **does a job and does it well**
 - Some tasks needs using more than one tool
 - Eg: Print a sorted list of home directories of users whose shell is “/bin/bash”. We again use /etc/passwd here!
 - Step1: Divide the task into smaller pieces
 - Subtask 1: Get list of users whose shell is /bin/bash
 - Subtask 2: Sort the list
 - Step 2: Find commands that do the sub tasks
 - `grep`, `cut` and `sort`
 - Step 3: Combine and execute the commands --- **HOW?**

Combining Commands

- How to combine Unix commands
 - Most tools take input from STDIN if the file argument is missing
 - Redirections provided by shell
 - Output redirection “>” sends output from a command to a file
 - Input redirection “<” sends input to a command from a file
 - Using it in our scenario
 - `grep "/bin/bash" /etc/passwd > list_of_matching_lines.txt`
 - `cut -f6 -d: list_of_matching_lines.txt > list_of_users.txt`
 - `sort list_of_users.txt`

Combining Commands

– The 'pipe'

- The pipe “|” takes STDOUT of one command and feeds it as STDIN to another command
- Using it in our scenario

```
– grep “/bin/bash” /etc/passwd | cut -f6 -d: | sort
```

• What if you want to delete all the resulting directories?

- `rm`, but it needs the list to be in the same line not multiple lines as the above command outputs it!

– `xargs` is the solution

- `xargs` joins all the lines in its input and passes it to another command
- `grep “/bin/bash” /etc/passwd | cut -f6 -d: | sort | xargs rm -f`

Combining Commands

- The possibilities are unlimited!
- The grep command also takes a regular expression as a pattern to be searched for
- The command find, finds file but can do it based on any criteria you think of (modified times, expressions in names, type of files etc;)
- Simple, specialized and perfect – thats the unix way!

Managing your GNU/Linux

- GNU/Linux startup (every unix does it the same way)
 - Kernel starts up
 - `init` is the only process started by the kernel
 - `init` then parses `/etc/inittab` and accordingly starts the daemons
 - Most daemon startup scripts are present in `/etc/init.d`
 - GNU/Linux has various runlevels – 0 to 6
 - corresponding scripts are present in `/etc/rc<runlevel>.d` and are generally symlinks to files present in `/etc/init.d`
 - `init` grabs the terminals and starts shells on each of them

Managing your GNU/Linux

- Starting a daemon or not-starting it is as easy as adding or removing the symlink from `/etc/rc<runlevel>.d`
- So much about `/etc`?
 - It is where all the configuration files stay
 - The entire unix system (GNU/Linux here) can be controlled by modifying the settings here
- Specific details about configuring some of unix daemons in the sessions to follow.

Managing your GNU/Linux

- Runlevels
 - Configuration profiles for different sets of daemons and features of a unix system
 - 0: Shutdown
 - 1: Generally single-user and no network
 - 2: Full featured with selected daemons
 - 6: Reboot

Thanks!

s.prasad@gmail.com

The unix flavour image from <http://en.wikipedia.org/wiki/UNIX>