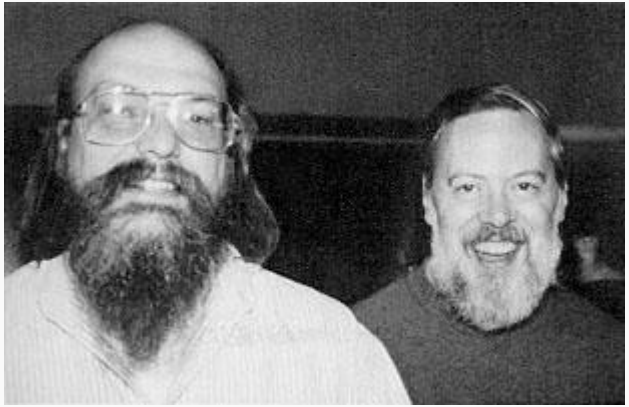
Unix and Linux

Gokcehan Kara <gokcehan.kara@boun.edu.tr>

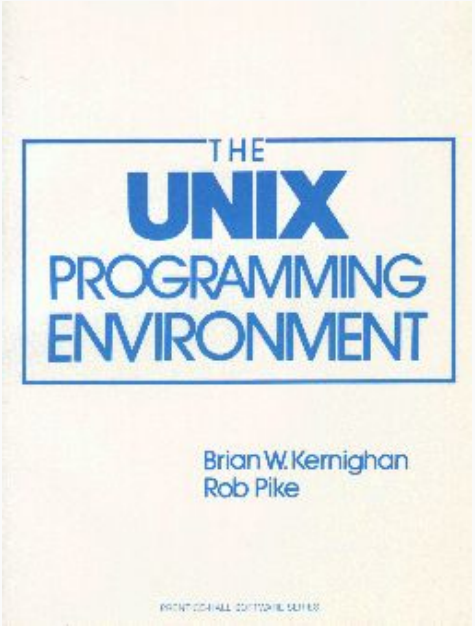
UNIX History

- Development dates back to 1960s at AT&T Bell Labs
- Multitasking and Multiuser OS
- First assembly version in 1969
- Rewritten in C in 1972
- Leads to numerous variants (both academical and commercial)
- Today, UNIX name usually refers to OS family, not just a single OS



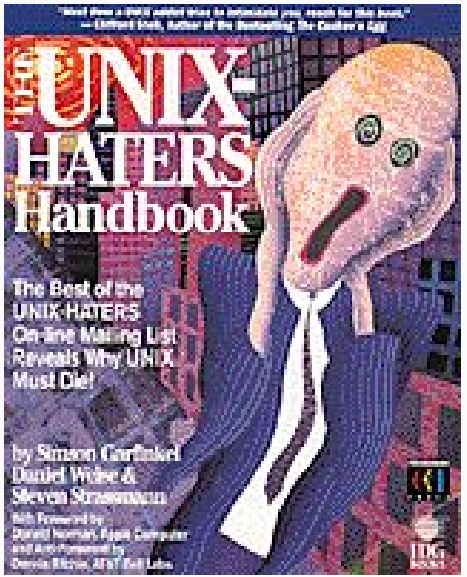
UNIX Legacies

- Simple tools (minimalism)
- Unix pipes (composability)
- Regular expressions
- Everything is a file

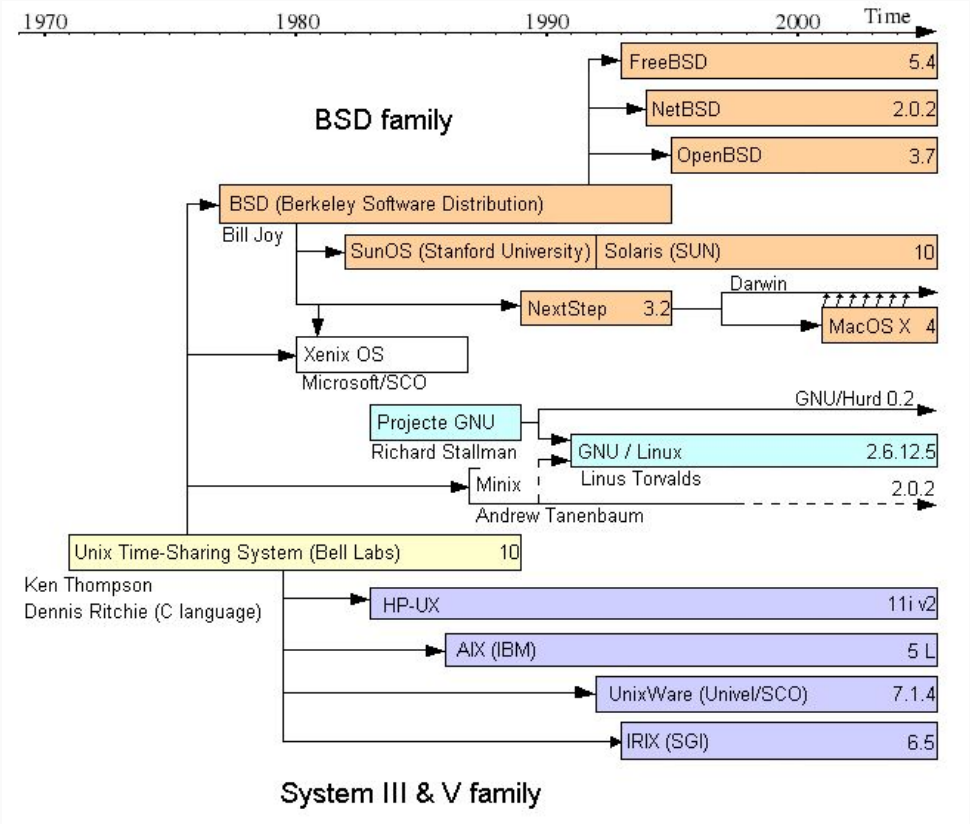


UNIX Pitfalls

1. Unix - The World's First Computer Virus
2. Welcome, New User! - Like Russian Roulette with Six Bullets Loaded
3. Documentation? - What documentation?
4. ...



UNIX Family


1970
1980
1990
2000
Time
BSD family
FreeBSD 5.4
NetBSD 2.0.2
OpenBSD 3.7
BSD (Berkeley Software Distribution)
Bill Joy
SunOS (Stanford University)
Solaris (SUN) 10
Darwin
NextStep 3.2
MacOS X 4
Xenix OS
Microsoft/SCO
GNU/Hurd 0.2
Projecte GNU
Richard Stallman
GNU / Linux 2.6.12.5
Linus Torvalds
Minix 2.0.2
Andrew Tanenbaum
Unix Time-Sharing System (Bell Labs) 10
Ken Thompson
Dennis Ritchie (C language)
HP-UX 11i v2
AIX (IBM) 5 L
UnixWare (Univel/SCO) 7.1.4
IRIX (SGI) 6.5
System III & V family

Documentaries

1. https://archive.org/details/UNIX1985
2. https://archive.org/details/RevolutionOS

OS Standards

- Portable Operating System Interface (POSIX) by IEEE (1988)
- Single UNIX Specification (SUS) by The Open Group (1990s)
- Later merged as Open Group Base Specification by Austin Group (2008)

POSIX for Windows

- Cygwin (https://www.cygwin.com/)
- MinGW (http://www.mingw.org/)
- Windows Subsystem for Linux (WSL) (Beta) (https://blogs.msdn.microsoft.com/wsl/)

GNU/Linux

GNU components include

- GCC
- GDB
- LibC
- CoreUtils
- BinUtils
- Build System
- Bash Shell
- Gnome Desktop

and Linux is the OS kernel

GNU/Linux Distributions

A software collection including

- Kernel (e.g. Linux)
- Userland (e.g. GNU)
- Window System (e.g. xorg)
- Window Manager (e.g. Compiz)
- Desktop Env. (e.g. GNOME)
- Package manager (e.g. dpkg)
- Software repositories
- Textual/Graphical installer

redhat

debian

CentOS

archlinux

openSUSE

ubuntu

Linux Mint

from freedom came elegance

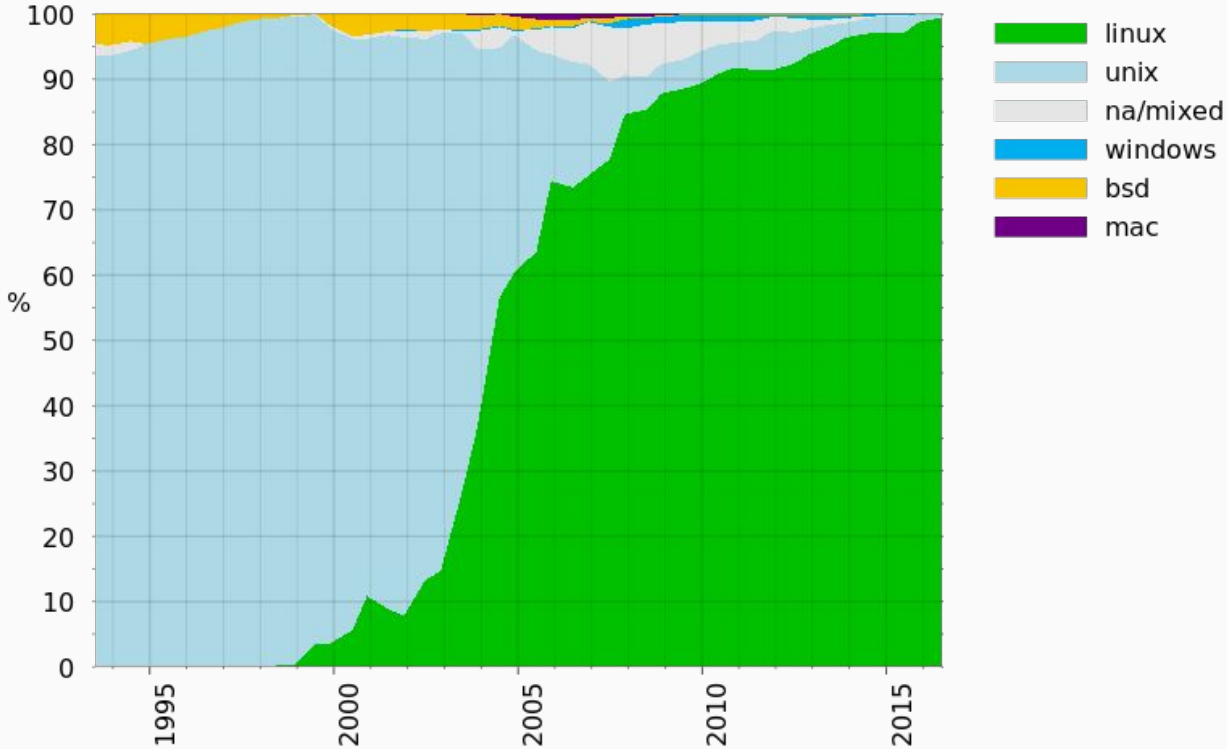
OS Usage Share (Desktop & Mobile)
(https://en.wikipedia.org/wiki/Usage_share_of_operating_systems)

Category	Source	Date	Linux	Unix and Unix-like (not incl. Linux)	Windows	Other
Desktop, laptop (excluding Android and Chrome OS)	Net Applications	August 2017	3.37% (Ubuntu, etc.)	5.94% (macOS)	90.70%	
Smartphone, tablet	StatCounter Global Stats	July 2017	70.27% (Android)	22.47% (iOS)	0.81%	6.43%

OS Usage Share (Web Servers)
(https://en.wikipedia.org/wiki/Usage_share_of_operating_systems)

Source	Date	Unix, Unix-like				Microsoft Windows
		All	Linux	FreeBSD	Unknown	
W3Techs	Feb 2015	67.80%	35.90%	0.95%	30.90%	32.30%
Security Space	Feb 2014	<79.3%	N/A			>20.7%
W3Cook	May 2015	98.30%	96.60%	1.70%	0%	1.70%

OS Usage Share (Supercomputers)
(https://en.wikipedia.org/wiki/Usage_share_of_operating_systems)



References

1. https://en.wikipedia.org/wiki/Unix
2. https://en.wikipedia.org/wiki/History_of_Unix
3. https://en.wikipedia.org/wiki/Unix_philosophy
4. https://en.wikipedia.org/wiki/Linux
5. https://en.wikipedia.org/wiki/GNU
6. https://en.wikipedia.org/wiki/GNU_Project
7. https://en.wikipedia.org/wiki/Linux_distribution
8. https://en.wikipedia.org/wiki/Desktop_environment

Shell

- User interface for OS services including:
  - File management
  - Process management
  - Task automation
  - System Monitoring
  - System Configuration
- Textual (e.g. cmd.exe or bourne shell) or Graphical (e.g. Windows or Unity)

Unix Shells

- Bourne shell (sh) (default in UNIX Version 7)
- Almquist shell (ash)
- Debian almquist shell (dash)
- Bourne-again shell (bash) (default in Linux and MacOS X)
- Korn shell (ksh)
- C shell (csh)
- Tenex C shell (tcsh) (default in FreeBSD)

Motivational Story (1986)

Jon Bentley asks Donald Knuth to demonstrate “literate programming” for his column named “Programming Pearls” in Communications of the ACM magazine. He asks for a solution to the following problem:

> “Given a text file and an integer k, print the k most common words in the file (and the number of their occurrences) in decreasing frequency.”

Knuth comes up with a program in WEB (a literate PASCAL system) that is more than 10 pages in length.

Motivational Story (1986) (Cont.)

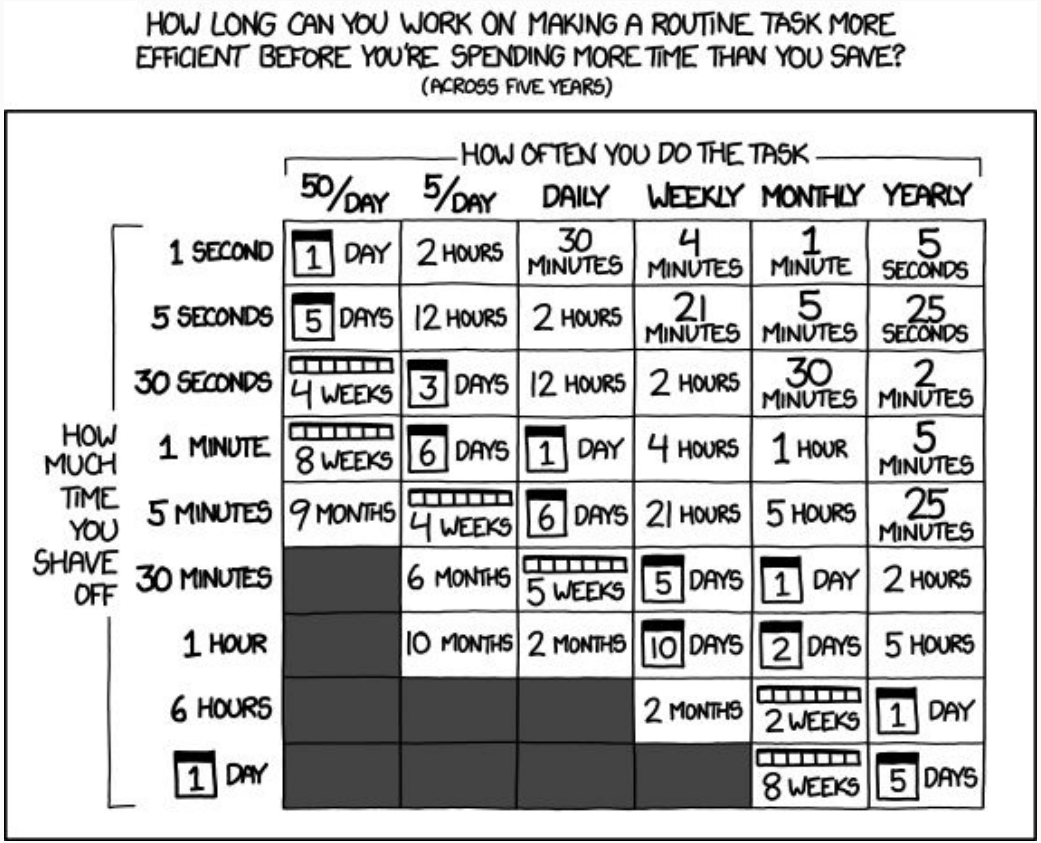
Next week, Jon Bentley asks Douglas McIlroy to review Knuth's program.

He reviews the code thoroughly and says it's brilliant but not wise. He then comes up with the following solution in shell:

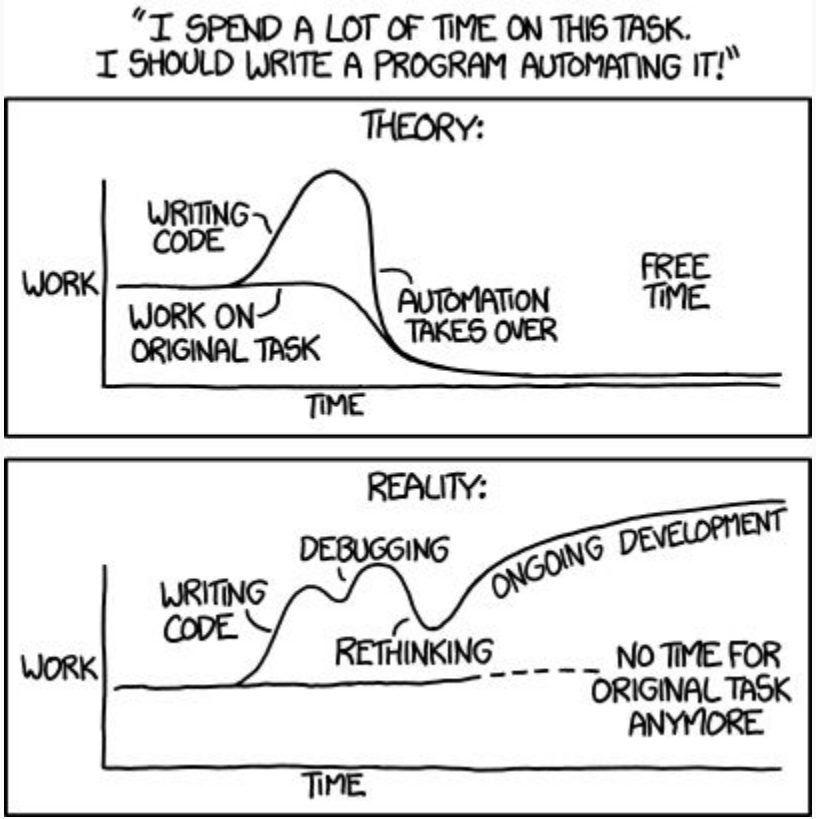
```
tr -cs A-Za-z '\n' | tr A-Z a-z | sort | uniq -c | sort -rn | sed ${1}q
```

Full story: http://dl.acm.org/citation.cfm?id=5948.315654

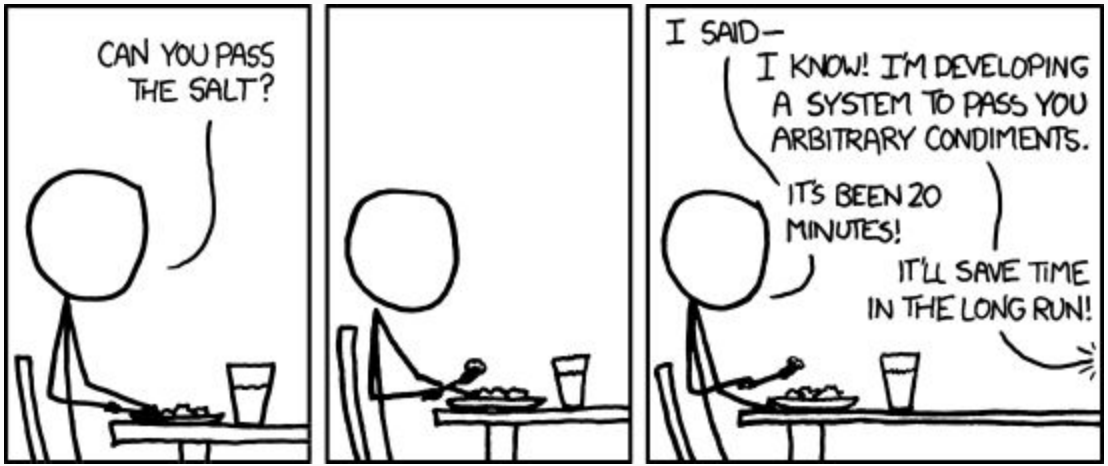
Is It Worth the Time? (https://xkcd.com/1205/)



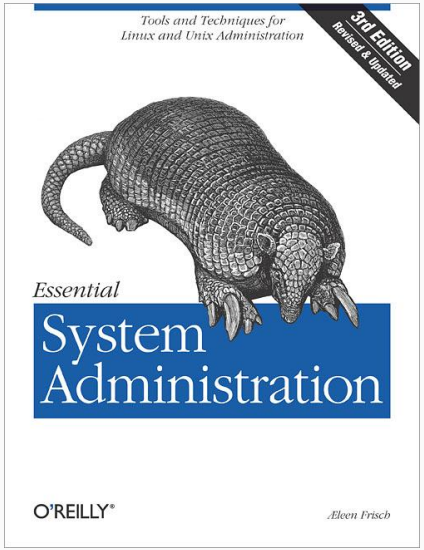
Automation (https://xkcd.com/1319/)

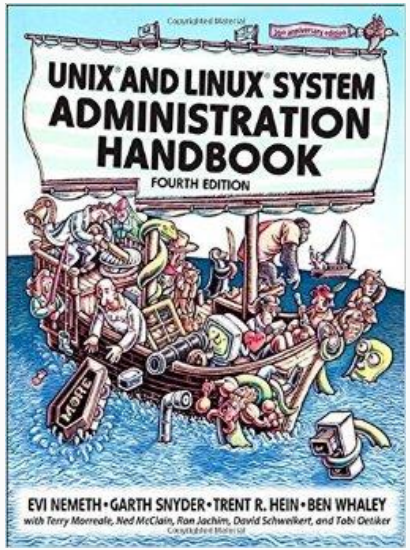


The General Problem (https://xkcd.com/974/)



Books


Tools and Techniques for
Linux and Unix Administration
3rd Edition
Revised & Updated
Essential
System
Administration
O'REILLY®
Æleen Frisch


UNIX® AND LINUX® SYSTEM
ADMINISTRATION
HANDBOOK
FOURTH EDITION
EVI NEMETH • GARTH SNYDER • TRENT R. HEIN • BEN WHALEY
with Terry Morreale, Ned McClain, Ran Jachim, David Schweikert, and Tobi Oetiker

Ubuntu on VirtualBox (demo)

ls (1) - list directory contents

Common options include:

- `-a` to not ignore entries starting with .
- `-l` to use a long listing format
- `-h` to print human readable sizes
- `-t` to sort by modification time, newest first
- `-S` to sort by file size, largest first
- `-r` to reverse order while sorting
- `-R` to list subdirectories recursively

Manuals and Documentations

- whatis (1) - display one-line manual page descriptions
- man (1) - an interface to the on-line reference manuals
- apropos (1) - search the manual page names and descriptions
- manpath (1) - determine search path for manual pages
- info (1) - read Info documents
- help (information about builtin commands) (shell builtin)
- type (information about command type) (shell builtin)

Man Sections (see `man man`)

1. Executable programs or shell commands
2. System calls (functions provided by the kernel)
3. Library calls (functions within program libraries)
4. Special files (usually found in /dev)
5. File formats and conventions eg /etc/passwd
6. Games
7. Miscellaneous (including macro packages and conventions)
8. System administration commands (usually only for root)
9. Kernel routines [Non standard]

Unix Paths

- Root directory is denoted with `/`
- Home directory is denoted with `~` (expanded by shell)
- Path separator is `/`
- Current directory is denoted with `.`
- Parent directory is denoted with `..`
- Absolute paths are those that start with `/`
- Relative paths are those that does not start with `/`

Path Navigation and File/Dir Manipulation

- pwd (1) - print name of current/working directory
- cd (change directory) (shell builtin)
- mkdir (1) - make directories
- rmdir (1) - remove empty directories
- touch (1) - change file timestamps
- rm (1) - remove files or directories
- cp (1) - copy files and directories
- mv (1) - move (rename) files
- ln (1) - make links between files

Exercise

- Examine the file system (see `man hier`)
- Create the following directory structure and files:

```
/home/gokce/Desktop/
└── demo/
    ├── bar.txt
    └── foo.txt

1 directory, 2 files
```

File Displaying/Editing

- file (1) - determine file type
- cat (1) - concatenate files and print on the standard output
- head (1) - output the first part of files
- tail (1) - output the last part of files
- more (1) - file perusal filter for crt viewing
- less (1) - opposite of more
- gedit (1) - text editor for the GNOME Desktop
- nano (1) - Nano's ANOther editor, an enhanced free Pico clone
- vi (1) - Vi IMproved, a programmers text editor

Exercise

Add some content to the files you created before and then display them

Pipes and Redirection

- Every process has at least 3 communication channels
  - Standard Input (STDIN)
  - Standard Output (STDOUT)
  - Standard Error (STDERR)
- Unix has file descriptors for these channels as 0, 1, and 2
- Most commands reads from STDIN and outputs to STDOUT conventionally
- `<`, `>`, and `>>` are used to redirect these channels from/to files
- `|` is used to connect STDOUT of one process to STDIN of another
- tee (1) - read from standard input and write to standard output and files

Exercise

- Create a file with a list of folders under root
- Print k th line in a file using `head` and `tail`
- Create random strings from `/dev/urandom` using `strings`

echo (1) - display a line of text

printf (1) - format and print data

strings (1) - print the strings of printable characters in files.

Special Device Files

- `/dev/null` accepts and discards all input (returns EOF)
- `/dev/zero` accepts and discards all input (returns NUL)
- `/dev/full` returns NUL bytes when read and ENOSPC when written
- `/dev/random` and `/dev/urandom` produces pseudo-random numbers
- `/dev/tty` refers to the underlying terminal
- `/dev/stdin` refers to the STDIN of the current process
- /dev/stdout` refers to the STDOUT of the current process
- /dev/stderr` refers to the STDERR of the current process

Exercise

Send error messages to null device

Users

- Could be real (e.g. a person) or pseudo (e.g. a service)
- Determined by a User Identification Number (UID)
- Accounts stored in `/etc/passwd`
- Passwords stored in `/etc/shadow`
- su (1) - change user ID or become superuser
- sudo (8) - execute a command as another user

User Management

- users (1) - print the user names of users currently logged in
- w (1) - Show who is logged on and what they are doing.
- who (1) - show who is logged on
- whoami (1) - print effective userid
- useradd (8) - create a new user or update default new user information
- usermod (8) - modify a user account
- userdel (8) - delete a user account and related files
- passwd (1) - change user password
- chsh (1) - change login shell

Exercise

Create a new user, change its password and set an appropriate login shell

Groups

- Determined by a Group Identification Number (GID)
- A user can belong to multiple groups
- Each user has also a group with the same name as the user name
- Groups stored in `/etc/group`
- Group passwords stored in `/etc/gshadow`

Group Management

- id (1) - print real and effective user and group IDs
- groups (1) - print the groups a user is in
- groupadd (8) - create a new group
- groupmod (8) - modify a group definition on the system
- groupdel (8) - delete a group

Exercise

Create a new group (e.g. `cmpe`) and add the previously created user to it

Files

- Everything is a file (UNIX legacy)
- Commands are executable files
- System privilege and permissions are controlled with files
- Device I/O and File I/O uses files (except at the lowest level)
- Interprocess communication uses files (UNIX pipes)
- All disks are combined as a single tree under root (denoted with `/`)
- File accesses are handled with ownership and protection
- Each file has an owner user and an owner group

File Permissions

- First bit denotes the type (e.g. file/dir)
- For each file there are 3 access sets
  - User access (u)
  - Group access (g)
  - Other access (o)
- For each access set there are 3 access types
  - Read (r)
  - Write (w)
  - Execute (x)
- In total there are 9 permissions bits
- There are 3 more bits reserved for special purposes (setuid, setgid, sticky)

File Ownership

- chown (1) - change file owner and group
- chgrp (1) - change group ownership
- chmod (1) - change file mode bits
- umask - display or set file mode mask (shell builtin)

Exercise

- What is the numeric file mode equivalent of the following accesses?
  - `rwxr-xr--`
  - `--x-w--wx`
  - `-r-rw-x-w` (is there anything wrong with this example?)
- What is the accesses equivalent of the following numeric file modes?
  - 755
  - 644
  - 325
- Change the owner and/or group of files/dirs and try permissions.

Searching

- grep (1) - print lines matching a pattern
- which (1) - locate a command
- find (1) - search for files in a directory hierarchy
- locate (1) - find files by name
- whereis (1) - locate the binary, source, and manual page files for a command

Exercise

Try search commands

Commands

A command could be:

- An executable file in `$PATH`
- A shell builtin (implemented either as necessity or performance)
- An alias (defined with `alias` builtin)
- Relative/Absolute path to an executable file

Cat (w/o args) Command (K&R)

```
#include <stdio.h>

main() {
    int c;
    while ((c = getchar()) != EOF)
        putchar(c);
}
```

Echo Command (K&R)

```
#include <stdio.h>

main(int argc, char *argv[]) {
    while (--argc > 0)
        printf("%s%s", *++argv, (argc > 1) ? " " : "");
    printf("\n");
}
```

Exercise

Echo the words `hello` and `world` with more than one space between them

Packages

- Software distribution in archive files
- Collected in repositories by distributions
- Comes as either pre-built binary (usually) or source code forms
- Handled by package managers:
  - Installation
  - Upgrading
  - Configuring
  - Removal

Package Managers

- apt (8) - command-line interface
- apt-get (8) - APT package handling utility - - command-line interface
- apt-cache (8) - query the APT cache
- dpkg (1) - package manager for Debian
- rpm (8) - RPM Package Manager
- yum (8) - Yellowdog Updater Modified

Exercise

Do an example for each of the following:

- Show info of a package
- Install a package
- Remove a package (with dependencies)
- Update list of packages
- Upgrade packages
- List installed files of a package
- Find the owner package of a file

Manual Installation

- A package might not be available in the distro’s repositories
- The package version in the repositories might be outdated
- Traditionally installed under `/usr/local` or `/opt`
- Most packages use the convention `./configure; make; make install`

Exercise

Install a package manually

tar (1) - The GNU version of the tar archiving utility

unzip (1) - list, test and extract compressed files in a ZIP archive

unrar-nonfree (1) - extract files from rar archives

7z (1) - A file archiver with highest compression ratio

Processes

- Single executable running in its own address space
- Unix jobs or commands may be composed of multiple processes
- Interactive processes can run in either the background or the foreground
  - `&` is used to run a command in the background
  - ^Z stops foreground process
  - ^C kill foreground process
- Batch processes are submitted to a queue for execution
- Daemons are server processes running continuously while system is up

Process Attributes

- Process ID (PID)
- Parent process ID (PPID)
- Nice number
- TTY
- Real and effective user ID (RUID, EUID)
- Real and effective group ID (RGID, EGID)

Process Manipulation/Monitoring

- jobs - lists the active jobs (shell builtin)
- fg - move job to the background (shell builtin)
- bg - move job to the foreground (shell builtin)
- disown - remove jobs from current shell (shell builtin)
- ps (1) - report a snapshot of the current processes
- pstree (1) - display a tree of processes
- pgrep (1) - look up or signal processes based on name and other attributes
- top (1) - display Linux processes

Exercise

- Start a process in the foreground and then move to background
- Find a way to make the process continue after closing the terminal
- List all processes of a user and sort by
  - Cpu usage
  - Memory usage

Process Signals

- signal (7) - overview of signals
- kill (1) - send a signal to a process
- killall (1) - kill processes by name
- pkill (1) - look up or signal processes based on name and other attributes

Exercise

Create a cpu intensive process and then kill it

Networking

- ping (8) - send ICMP ECHO_REQUEST to network hosts
- hostname (1) - show or set the system's host name
- arp (8) - manipulate the system ARP cache
- inetutils-traceroute (1) - Trace the route to a host
- netstat (8) - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
- tcpdump (8) - dump traffic on a network

Exercise

Explore your network with networking tools

Network Configuration

- ifconfig (8) - configure a network interface
- ifup (8) - bring a network interface up
- ifdown (8) - take a network interface down
- ip (8) - show / manipulate routing, devices, policy routing and tunnels

Exercise

- List all available network interfaces
- Restart an interface
- Find out your IP address
- Change MTU size
- Change your MAC address

Domain Name System (DNS)

- Translates domain names to IP addresses
- Known hosts are listed in `/etc/hosts`
- DNS servers are configured in `/etc/resolv.conf`

Exercise

- Query the address of a site
- Query the address of a site using a different DNS server
- Reverse query an IP address

nslookup (1) - query Internet name servers interactively

host (1) - DNS lookup utility

dig (1) - DNS lookup utility

Remote Connections

- telnet (1) - user interface to the TELNET protocol
- nc (1) - arbitrary TCP and UDP connections and listens
- ssh (1) - OpenSSH SSH client (remote login program)
- scp (1) - secure copy (remote file copy program)
- rsync (1) - a fast, versatile, remote (and local) file-copying tool
- sshfs (1) - filesystem client based on ssh
- curl (1) - transfer a URL
- wget (1) - The non-interactive network downloader.

Exercise

- Make a GET request from a web server
- Test an SMTP mail server
- Transfer a file using TCP
- Connect to a remote server using SSH
- Transfer a file using SSH
- Download a page from internet
- Download a file from internet

Links

1. http://www.tldp.org/
2. https://www.die.net/
3. https://linuxjourney.com
4. http://www.commandlinefu.com
5. https://wiki.ubuntu.com/DashAsBinSh
6. http://mywiki.wooledge.org/BashPitfalls