

Networks & Server Structures

Binary

Data Storage Units

Terminal Commands

File Permissions

Decimal System

1000^m	10^n	Prefix	Symbol	Since ^[1]	Short scale	Long scale	Decimal
1000^8	10^{24}	yotta-	Y	1991	Septillion	Quadrillion	1 000 000 000 000 000 000 000 000
1000^7	10^{21}	zetta-	Z	1991	Sextillion	Trilliard	1 000 000 000 000 000 000 000
1000^6	10^{18}	exa-	E	1975	Quintillion	Trillion	1 000 000 000 000 000 000
1000^5	10^{15}	peta-	P	1975	Quadrillion	Billiard	1 000 000 000 000 000
1000^4	10^{12}	tera-	T	1960	Trillion	Billion	1 000 000 000 000
1000^3	10^9	giga-	G	1960	Billion	Milliard	1 000 000 000
1000^2	10^6	mega-	M	1960	Million		1 000 000
1000^1	10^3	kilo-	k	1795	Thousand		1 000
$1000^{2/3}$	10^2	hecto-	h	1795	Hundred		100
$1000^{1/3}$	10^1	deca-	da	1795	Ten		10
1000^0	10^0	(none)	(none)	NA	One		1

Binary System

- ✦ Base 2 system
- ✦ Consists of only 1's and 0's
- ✦ Concept possibly as old as the 8th century BC
- ✦ Modern binary system documented by Gottfried Leibniz

Binary System

- ✧ Bit Values

- ✧ 0 0 0 0 0 0 0 0

- ✧ 128 64 32 16 8 4 2 1

Binary System

- ✦ Bit Values

- ✦ 1 0 0 0 1 0 0 1

- ✦ 128 64 32 16 8 4 2 1

- ✦ $10001001 = 128 + 8 + 1 = 137$

Practice

✖ 10101010

✖ 11101101

✖ 01000001

✖ 00100111

✖ 01001101

✖ 00110111

✖ 00001101

✖ 01011110

✖ 10010011

✖ 11100010

✖ 01010101

✖ 11110000

✖ 10110001

✖ 11111111

Practice

✧ 10101010 - 170

✧ 11101101 - 237

✧ 01000001 - 65

✧ 00100111 - 39

✧ 01001101 - 77

✧ 00110111 - 55

✧ 00001101 - 13

✧ 01011110 - 94

✧ 10010011 - 147

✧ 11100010 - 226

✧ 01010101 - 85

✧ 11110000 - 240

✧ 10110001 - 177

✧ 11111111 - 255

Practice

✦ 63

✦ 99

✦ 72

✦ 15

✦ 18

✦ 56

✦ 81

✦ 182

✦ 46

✦ 27

✦ 169

✦ 200

Practice

- ✧ 63 - 00111111
- ✧ 99 - 01100011
- ✧ 72 - 01001000
- ✧ 15 - 00001111
- ✧ 18 - 00010010
- ✧ 56 - 00111000
- ✧ 81 - 01010001
- ✧ 182 - 10110110
- ✧ 46 - 00101110
- ✧ 27 - 00011011
- ✧ 169 - 10101001
- ✧ 200 - 11001000

Data Storage Units

- ✦ Units for representing data size
- ✦ Bit
- ✦ Byte = 8 bits
- ✦ Kilobyte (KB) = 1000 bytes
- ✦ Megabyte (MB) = 1000 Kilobytes
- ✦ Gigabyte (GB) = 1000 Megabytes
- ✦ Terabyte (TB) = 1000 Gigabytes
- ✦ Petabyte (PB) = 1000 Terabytes

Data Storage Units

- ✧ BI Prefix
- ✧ Kibibyte = 1024 bytes = KiB
- ✧ Mebibyte = 1024 KiB = MiB
- ✧ Gibibyte = 1024 MiB = GiB
- ✧ Tebibyte = 1024 GiB = TiB

How big is a Yottabyte?

TERABYTE

Will fit 200,000 photos or mp3 songs on a single 1 terabyte hard drive.



PETABYTE

Will fit on 16 Backblaze storage pods racked in two datacenter cabinets.



EXABYTE

Will fit in 2,000 cabinets and fill a 4 story datacenter that takes up a city block.



ZETTABYTE

Will fill 1,000 datacenters or about 20% of Manhattan, New York.



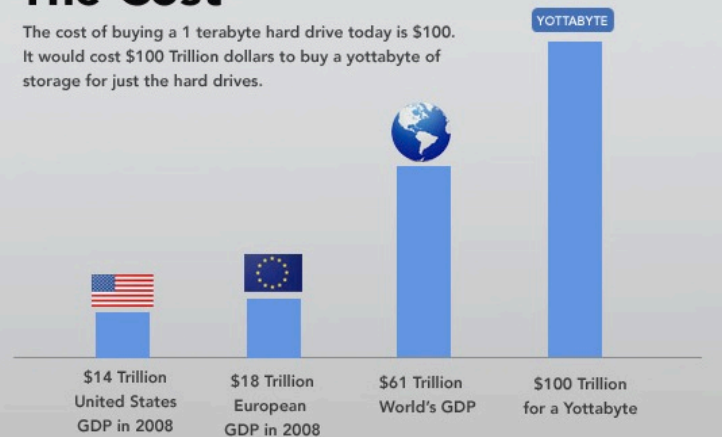
YOTTABYTE

Will fill the states of Delaware and Rhode Island with a million datacenters.



The Cost

The cost of buying a 1 terabyte hard drive today is \$100. It would cost \$100 Trillion dollars to buy a yottabyte of storage for just the hard drives.



Bandwidth Notation

- ✧ Kbps = kilobits per second
- ✧ KBps = kilobytes per second
- ✧ Mbps = megabits per second
- ✧ MBps = megabytes per second
- ✧ Gbps = gigabits per second
- ✧ GBps = gigabytes per second

Hexadecimal

- ✦ A Base 16 system
- ✦ Uses 0-9 and A-F
- ✦ A=10, B=11, C=12, D=13, E=14, F=15
- ✦ 4096 256 16 1
- ✦ 0x11AB = 0X11AB = 11ABh = \$11AB

Practice

- 0x12EF
- 0x342
- 0xE59
- 0x2F
- 0x20A
- 0xFF
- 0x200
- 0xF0
- 0x400
- 0x142
- 0x29A
- 0x1E

Practice

- 0x12EF - 4847
- 0x342 - 834
- 0xE59 - 3673
- 0x2F - 47
- 0x20A - 522
- 0xFF - 255
- 0x200 - 512
- 0xF0 - 240
- 0x400 - 1024
- 0x142 - 322
- 0x29A - 666
- 0x1E - 30



File Systems

& Directory Structures

File Systems

- ✦ Method for storing and organizing computer files and the data they contain to make it easy to find and access them
- ✦ Uses data storage devices such as hard drives, CD's and USB keys
- ✦ Maintain physical location of data on storage media

File Systems

- ✧ 2 Types
 - ✧ Hierarchical
 - ✧ Contains subdirectories
 - ✧ Flat
 - ✧ No subdirectories
 - ✧ Everything at root level
 - ✧ Inefficient

Metadata

- ✦ Data about data
- ✦ Very useful in searches

Common Disk File Systems

- ✧ FAT 32
- ✧ NTFS
- ✧ HFS+
- ✧ ext3
- ✧ ext4

FAT 32

- ✦ File Allocation Table
- ✦ Developed by Microsoft
- ✦ Introduced with Windows 95
- ✦ Maximum file size is 4GB
- ✦ Most compatible file system

NTFS

- ✦ New Technology File System
- ✦ Created by Microsoft using concepts from HPFS (IBM/Microsoft)
- ✦ Introduced with Windows NT
- ✦ 16 TB maximum file size
- ✦ Less compatible than FAT32
- ✦ Journaling - logs changes, preventing corruption

HFS+

- ✦ Hierarchical File System
- ✦ Developed by Apple Inc.
- ✦ 8 EB maximum file size
- ✦ Used in Mac OS X
- ✦ Supports Journaling

ext3

- ✦ Third Extended File System
- ✦ Development lead by Dr. Stephen C. Tweedie
- ✦ Used by most Linux distributions
- ✦ Maximum file size is 2TB
- ✦ Supports Journaling

ext4

- ✦ Fourth Extended File System
- ✦ Used by Linux distributions
- ✦ Maximum file size is 16TB
- ✦ Backwards compatible with ext3

Network File Systems

- ✦ NFS - Network File System
 - ✦ Allows a user on a client computer to access files over a network in a manner similar to how local storage is accessed
- ✦ SMB - Server Message Block
 - ✦ Provides shared access to files, printers, serial ports, and miscellaneous communications between nodes on a network
- ✦ AFP - Apple Filing Protocol

Directory Structure

- ✧ Folder
 - ✧ Subfolder
 - ✧ File
 - ✧ File
 - ✧ Subfolder
 - ✧ File
 - ✧ Subfolder

Root Directory

- ✦ The first or top-most directory in a hierarchy
- ✦ The starting point where all the branches originate

Windows Directory Structure

- ✧ *Drive*
 - ✧ eg; C:\
- ✧ Physical & virtual drives are named by letter therefore there is no “*formal*” root directory for the system
- ✧ Independent ‘root’ directories on each drive

Linux Directory Structure

- ✧ Filesystem Hierarchy Standard
 - ✧ Used by Unix, Linux, SunOS, Solaris, Mac OS X

✧ /

- ✧ **bin** - contains programs/scripts available to all users
- ✧ **dev** - (virtual directory) represents hardware devices
- ✧ **etc** - system configuration files
- ✧ **home** - user accounts and subdirectories
- ✧ **lib** - shared libraries (PERL, Python, C+) Files shared by different programs
- ✧ **mnt** - mounted drives (hard drives, CD drive, USB drive)
- ✧ **proc** - (virtual directory) system information; processes, pci info, bus info
- ✧ **root** - home folder for root user (not in /home for security reasons)
- ✧ **sbin** - programs requiring administrator level privileges; not for normal users
- ✧ **tmp** - stores temporary information; typically emptied on reboot
- ✧ **usr** - [Unix System Resources] - shared data, libraries, binaries, programs
- ✧ **var** - [Variable data] - log files, databases, web server files
 - ✧ **www** - web pages placed here if running a web server

Directory Structure

- = current directory
- .. = parent directory

Directory Structure

- ✦ The location of a file or a directory is called the path
- ✦ Absolute path
 - ✦ Begins at root
 - ✦ `/home/username`
- ✦ Relative path
 - ✦ Will look wherever you are at in the file structure
 - ✦ `./username`


```
-rw-r--r-- 1 root    root      5660 Jul 25 09:43 bash-4.0_p28.ebuild
drwxr-xr-x 2 portage portage 2048 May 30 03:35 files
-rw-r--r-- 1 portage portage  468 Feb  9 04:35 metadata.xml
mars@marsmain /usr/portage/app-shells/bash $ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
<herd>base-system</herd>
<use>
  <flag name='bashlogger'>Log ALL commands typed into bash; should ONLY be
    used in restricted environments such as honeypots</flag>
  <flag name='net'>Enable /dev/tcp/host/port redirection</flag>
  <flag name='plugins'>Add support for loading builtins at runtime via
    'enable'</flag>
</use>
</pkgmetadata>
mars@marsmain /usr/portage/app-shells/bash $ sudo /etc/init.d/bluetooth status
Password:
* status: started
mars@marsmain /usr/portage/app-shells/bash $ ping -q -c1 en.wikipedia.org
PING rr.esams.wikimedia.org (91.198.174.2) 56(84) bytes of data.

--- rr.esams.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
```

Terminal

...because GUI's are for sissies.

CLI

- ✦ CLI - Command Line Interface
- ✦ Interface that allows you to talk directly to your computer through typing commands without going through a GUI
- ✦ Different operating systems have different CLI's.
 - ✦ Xterm, Terminal, Konsole, Cygwin, Command Prompt, etc

Why Use a Command Line?

- ✦ powerful
- ✦ can be faster than GUI
- ✦ Can string together multiple utility programs into one line of input

So why do I have to learn it?

- ✦ Most web servers run Linux
- ✦ Most of these servers do not have a GUI
- ✦ As web professionals you will need to know how to interact with these servers
- ✦ Later in the program you will use CLI to write code and interact with different programs such as MySQL, Coldfusion, and ActionScript

CLI Syntax

- ✦ Syntax - the grammar that the commands must follow
 - ✦ Different between different operating systems
- ✦ Commands usually follow one of the following forms:
 - ✦ [doSomething] [how] [toFiles]
 - ✦ [doSomething] [how] [sourceFile] [destinationFile]
 - ✦ [doSomething] [how] < [inputFile] > [outputFile]
 - ✦ [doSomething] [how] | [doSomething] [how] | [do Something] [how] > [outputFile]

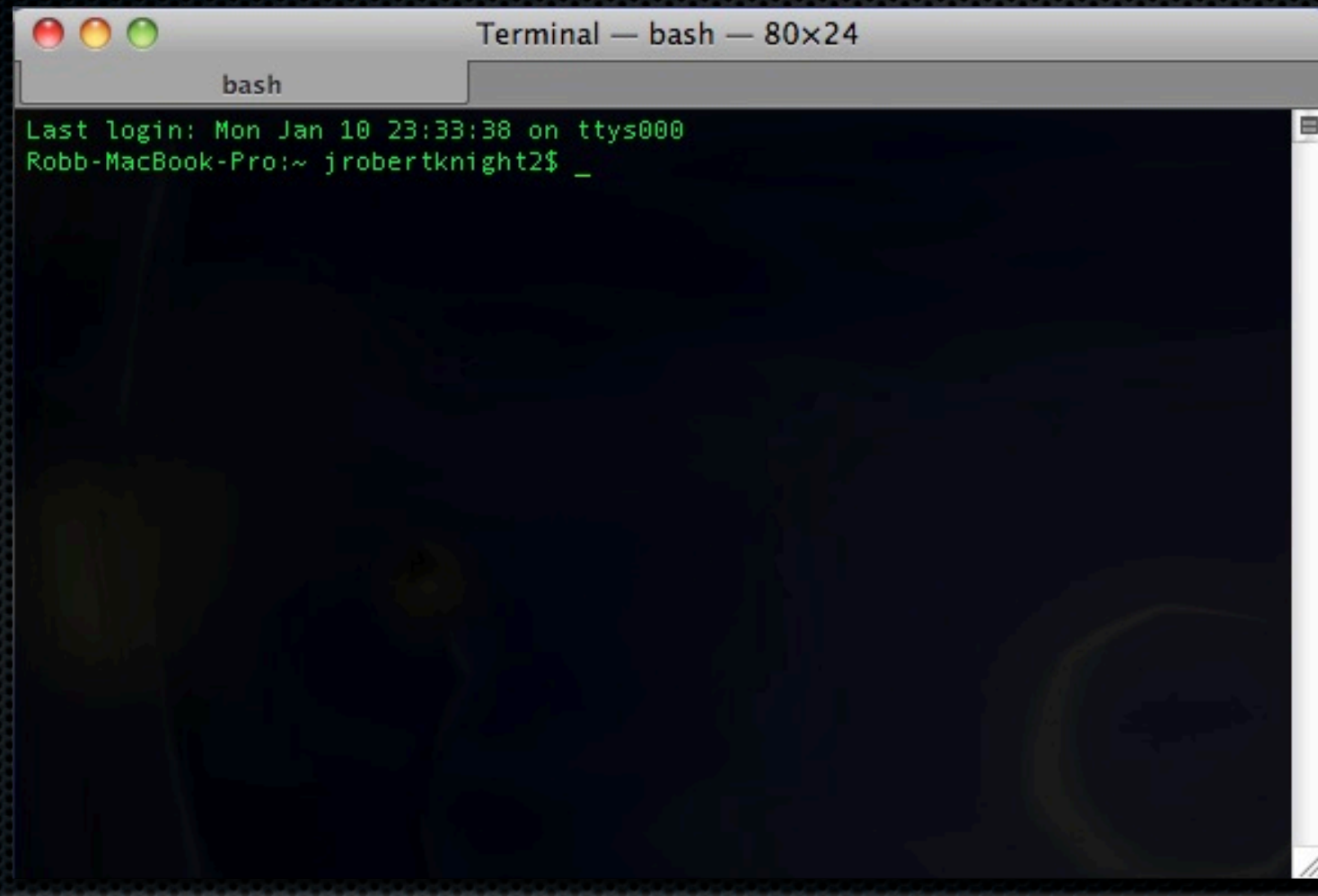
CLI

- ✦ Remember!!!
 - ✦ The command line does not treat you like an idiot. It assumes that you know what you are doing.
 - ✦ Case sensitivity matters!!!

Terminal

- ✦ Mac OS X command line utility
- ✦ Bash shell
 - ✦ **B**ourne **A**gain **S**hell
 - ✦ Designed as an open source clone of the Bourne shell (sh)

Terminal



Terminal Commands

- ✧ **pwd** - print working directory
 - ✧ Shows your current location in the file structure
- ✧ **ls** - list
 - ✧ Lists the contents of a directory
- ✧ **cd** - change directory
 - ✧ Moves you to another directory in the file structure

Terminal Commands

- ✧ **cp** - copy
 - ✧ Copies a file
- ✧ **mkdir** - make directory
 - ✧ Creates a new directory
- ✧ **rm** - remove / delete
 - ✧ Self explanatory. Use with caution.

Terminal Commands

- ✧ **mv** - move / rename
 - ✧ Trickiest command to work with
 - ✧ In some cases it moves a file or a directory
 - ✧ In other cases it renames a file or a directory
- ✧ **man** - manual
 - ✧ Displays a help file for a command
 - ✧ Type **q** (quit) to exit the manual

Terminal Commands

- ✧ Options

- ✧ Change or limit the way a command is executed
- ✧ Usually preceded with a dash “-”

- ✧ Examples

- ✧ **ls -l** - list contents of a directory in long format
- ✧ **ls -a** - includes hidden system files
- ✧ **ls -R** - recursive list. Lists everything in the directory and everything in any subdirectories

Terminal Commands

- ✦ **ping** - tests that a connection can be established between two nodes on a network
 - ✦ Used for troubleshooting

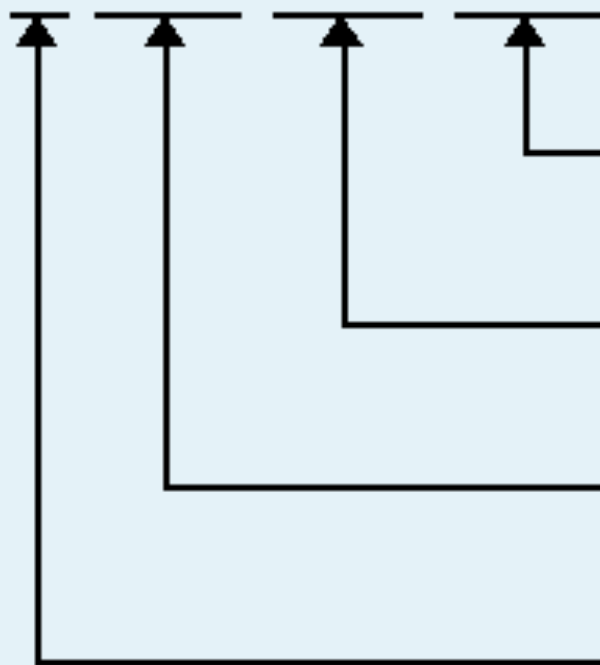
File Permissions

```
-rw----- 1 bshotts bshotts      576 Apr 17 1998 weather.txt
drwxr-xr-x 6 bshotts bshotts     1024 Oct  9 1999 web_page
-rw-rw-r-- 1 bshotts bshotts   276480 Feb 11 20:41 web_site.tar
-rw----- 1 bshotts bshotts      5743 Dec 16 1998 xmas_file.txt
```

File Name	Modification Time	Size (in bytes)	Group	Owner	File Permissions
weather.txt	Apr 17 1998	576	bshotts	bshotts	-rw-----
web_page	Oct 9 1999	1024	bshotts	bshotts	drwxr-xr-x
web_site.tar	Feb 11 20:41	276480	bshotts	bshotts	-rw-rw-r--
xmas_file.txt	Dec 16 1998	5743	bshotts	bshotts	-rw-----

File Permissions

`- rwxrw - r - -`



Read, write, and execute permissions
for all other users

Read, write and execute permissions
for members of the group owning the
file.

Read, write and execute permissions
for the owner of the file.

File type. "-" indicates a regular file. A
"d" indicates a directory.

File Permissions

```
rwX  rwX  rwX  =  111  111  111
rw-  rw-  rw-  =  110  110  110
rwX  ---  ---  =  111  000  000
```

and so on...

```
rwX  =  111  in binary =  7
rw-  =  110  in binary =  6
r-x  =  101  in binary =  5
r--  =  100  in binary =  4
```


File Permissions

Value	Meaning
777	(rwxrwxrwx) No restrictions on permissions. Anybody may do anything. Generally not a desirable setting.
755	(rwxr-xr-x) The file's owner may read, write, and execute the file. All others may read and execute the file. This setting is common for programs that are used by all users.
700	(rwx-----) The file's owner may read, write, and execute the file. Nobody else has any rights. This setting is useful for programs that only the owner may use and must be kept private from others.
666	(rw-rw-rw-) All users may read and write the file.
644	(rw-r--r--) The owner may read and write a file, while all others may only read the file. A common setting for data files that everybody may read, but only the owner may change.
600	(rw-----) The owner may read and write a file. All others have no rights. A common setting for data files that the owner wants to keep private.

File Permissions

Value	Meaning
777	(rwxrwxrwx) No restrictions on permissions. Anybody may list files, create new files in the directory and delete files in the directory. Generally not a good setting.
755	(rwxr-xr-x) The directory owner has full access. All others may list the directory, but cannot create files nor delete them. This setting is common for directories that you wish to share with other users.
700	(rwx-----) The directory owner has full access. Nobody else has any rights. This setting is useful for directories that only the owner may use and must be kept private from others.

chmod

- ✦ Used to change permissions on files or directories
- ✦ **chmod** [*permission*] [*filename*]
- ✦ Example
 - ✦ chmod 644 file2
 - ✦ Changes the permission for file 2 to -rw-r--r--

chmod

- ✧ Alternate Method
 - ✧ u = owner
 - ✧ g = group
 - ✧ o = others (everyone else)
 - ✧ a = all

chmod

- ✦ “+” adds the specified modes to the specified classes
- ✦ “-” removes the specified modes
- ✦ `chmod ug+rwx, o+r [file]`
- ✦ Sets permissions to `-rwxrwx-r--`

Terminal and Spaces

- ✦ It takes a little more effort to deal with file names and directories with spaces
 - ✦ Have to put the name in quotes
 - ✦ eg: “file 1” or “folder 1”
- ✦ This is why you often see file names with underscores in place of spaces
 - ✦ eg: file_1 or folder_1

FIN

Copyright Full Sail University

All rights are reserved by Full Sail University. Do not distribute, duplicate or otherwise alter this content without prior written consent of Full Sail University.