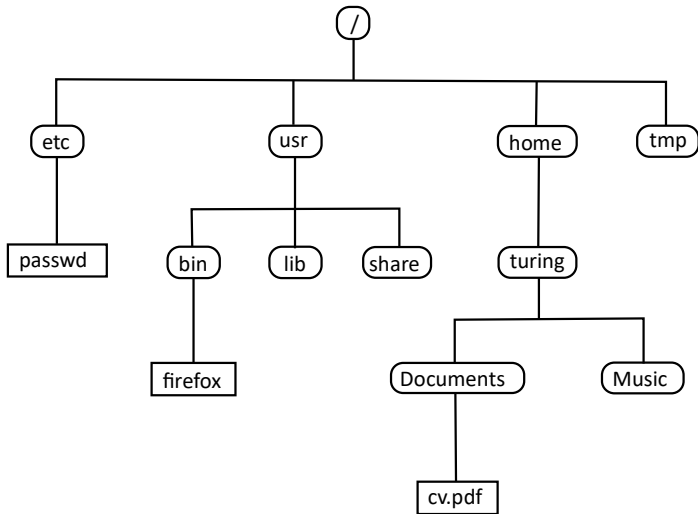


File System

- data and programs are kept on secondary storage
- conceptual unit: *file*
- a container of files: *folder*
- also called: *directory*
- it is also a file
- folders can be nested inside other folders
- top level folder: *root*

Unix File System



Paths

- how to refer to a file?
- *path*: sequence of folders, followed by the file
- absolute path: start from the root
- relative path: start from the “current” folder
- current folder: .
- parent folder (immediately above the current): ..

File Manager

- utility for file operations
- move around the file system
- create, delete
- copy, move, rename, ...

File Types

- two types of files
- *text*: (mostly) human-readable, easier to work with
- *binary*: only machine-readable, more efficient

File Extensions

- file names have an extension part
- for example: .pdf

MIME

- standard categorization of file types

<https://www.iana.org/assignments/media-types/media-types.xhtml>

- format: `type/subtype`
- types: image, audio, video, text, ...

MIME Examples

- image/jpeg, image/png
- audio/mpeg
- video/mp4, video/x-matroska
- application/pdf, application/zip
- text/html, text/plain

Text Editor

- application for creating and modifying text files

Archiving and Compression

- combine files and folders into one archive file
- compress a file for smaller file size
- extract archive file to get the original structure
- `tar` (archiving)
- `gzip`, `bzip2` (compression)
- `zip` (both)

User Interfaces

- how users interact with applications
- graphical environment: windows, mouse, ...
- *GUI*: graphical user interface
- terminal: type command, see result
- *CLI*: command line interface

Bistate Objects

- some objects are always in one of two states
- coins: heads/tails
- switches: on/off
- transistors: on/off

Binary Numbers

- computers represent information using *binary* numbers
- *bit*: binary digit
- one of two values: 0, 1

Representing Numbers

- digits correspond to powers of 2

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1

Binary Value Examples

decimal	binary
---------	--------

2	10
---	----

3	11
---	----

4	100
---	-----

5	101
---	-----

13	1101
----	------

22	10110
----	-------

Octets and Bytes

- 8 bits: *octet*
a *byte* is 8 bits (now)
- smallest unit of information
- values: [0..255]

Byte Value Examples

decimal	binary
---------	--------

0	00000000
---	----------

1	00000001
---	----------

22	00010110
----	----------

65	01000001
----	----------

128	10000000
-----	----------

255	11111111
-----	----------

Binary Value Notation

- is a value written in decimal or in binary?
- 101: is it 101 or 5?
- notation: binary values start with `0b`
- `0b101`

Larger Units

- 1 Kio (kibiocet) = 1024 octets
- 1 Mio (mebiocet) = 1024 Kio
- 1 Gio (gibiocet) = 1024 Mio
- 1 Tio (tebiocet) = 1024 Gio

- 1 kB (kilobyte) = 1024 bytes
- 1 MB (megabyte) = 1024 kB
- 1 GB (gigabyte) = 1024 MB
- 1 TB (terabyte) = 1024 GB

Hexadecimal Numbers

- reading binary numbers is difficult
- hexadecimal: base 16
- digits correspond to powers of 16

16^3	16^2	16^1	16^0
4096	256	16	1

Hexadecimal Digits

dec	bin	hex
-----	-----	-----

8	1000	8
---	------	---

9	1001	9
---	------	---

10	1010	A
----	------	---

11	1011	B
----	------	---

dec	bin	hex
-----	-----	-----

12	1100	C
----	------	---

13	1101	D
----	------	---

14	1110	E
----	------	---

15	1111	F
----	------	---

Hexadecimal Notation

- 1 hex digit: 4 bits
- 1 octet: 8 bits, 2 hexits (hex digits)
- notation: hex values start with 0x

Hexadecimal Value Examples

dec	bin	hex
-----	-----	-----

16	00010000	10
----	----------	----

30	00011110	1E
----	----------	----

255	11111111	FF
-----	----------	----

Hex-Binary Conversion

- match hexadecimal digits and groups of 4-bits

F	3	C	0
1111	0011	1100	0000

11110011111000000

110011110000000

0011	0011	1100	0000
3	3	C	0

Octal Numbers

- Not common
- octal: base 8
- digits correspond to powers of 8
- each octal digit is represented by 3 bits
- octal values start with 00

Character Sets

- how to represent letters, punctuation signs, ...?
- assign a number to each *character*
- a set of all such assignments: *character set*
- also called an “encoding”

ASCII Character Set

- 128 characters
- English letters
- digits
- punctuation signs
- special characters (some of them are commands)

ASCII Table

char	#
------	---

!	0x21
---	------

#	0x23
---	------

7	0x37
---	------

?	0x3F
---	------

@	0x40
---	------

char	#
------	---

A	0x41
---	------

B	0x42
---	------

Z	0x5A
---	------

a	0x61
---	------

z	0x7A
---	------

Case Sensitivity

- 'A' and 'a' have different numbers
- most programs consider these as different letters

ISO8859 Sets

- ASCII only for English
- use 8 bits per character: 256 characters
- ISO8859-1: Western European
- first 128 same as ASCII
- ISO8859-9: Turkish
- Turkish instead of Icelandic

ISO8859-1 and ISO8859-9

#	ISO8859-1	ISO8859-9
0x3F	?	?
0x41	A	A
0xC7	Ç	Ç
0xE7	ç	ç
0xD0	Ý	Ѓ
0xF0	ď	ђ

Unicode

- a way of encoding all characters in all writing systems
- UTF-32: 32 bits per character
- UTF-16: 16/24/32 bits per character
- UTF-8: 8/16/24/32 bits per character

UTF Examples

char	#
------	---

!	0x0021
---	--------

A	0x0041
---	--------

Ç	0x00C7
---	--------

∞	0x221E
---	--------

举	0x4E3E
---	--------

Metadata

- two types of data associated with a file
- actual data: content of the file
- *metadata*: data describing the content

Metadata Examples

- photo file:
 - actual data: photograph
 - metadata: who, where, when, image format, ...
- song file
 - actual data: song
 - metadata: title, artist, lyrics, audio format, ...

Text File Metadata

- actual data: text in the file
- metadata: copyright, author, ...
- *character set*

Providing Metadata

- in some file formats, metadata is stored in the file along with the actual data
- cameras store technical metadata in photograph file
- music files can contain title, artist, ...
- in some file formats, metadata has to be provided externally
- character set of a text file