university
of
tyumen

school
of advanced
studies

Information Technology - basic

Lecture 1
Computer hardware and software

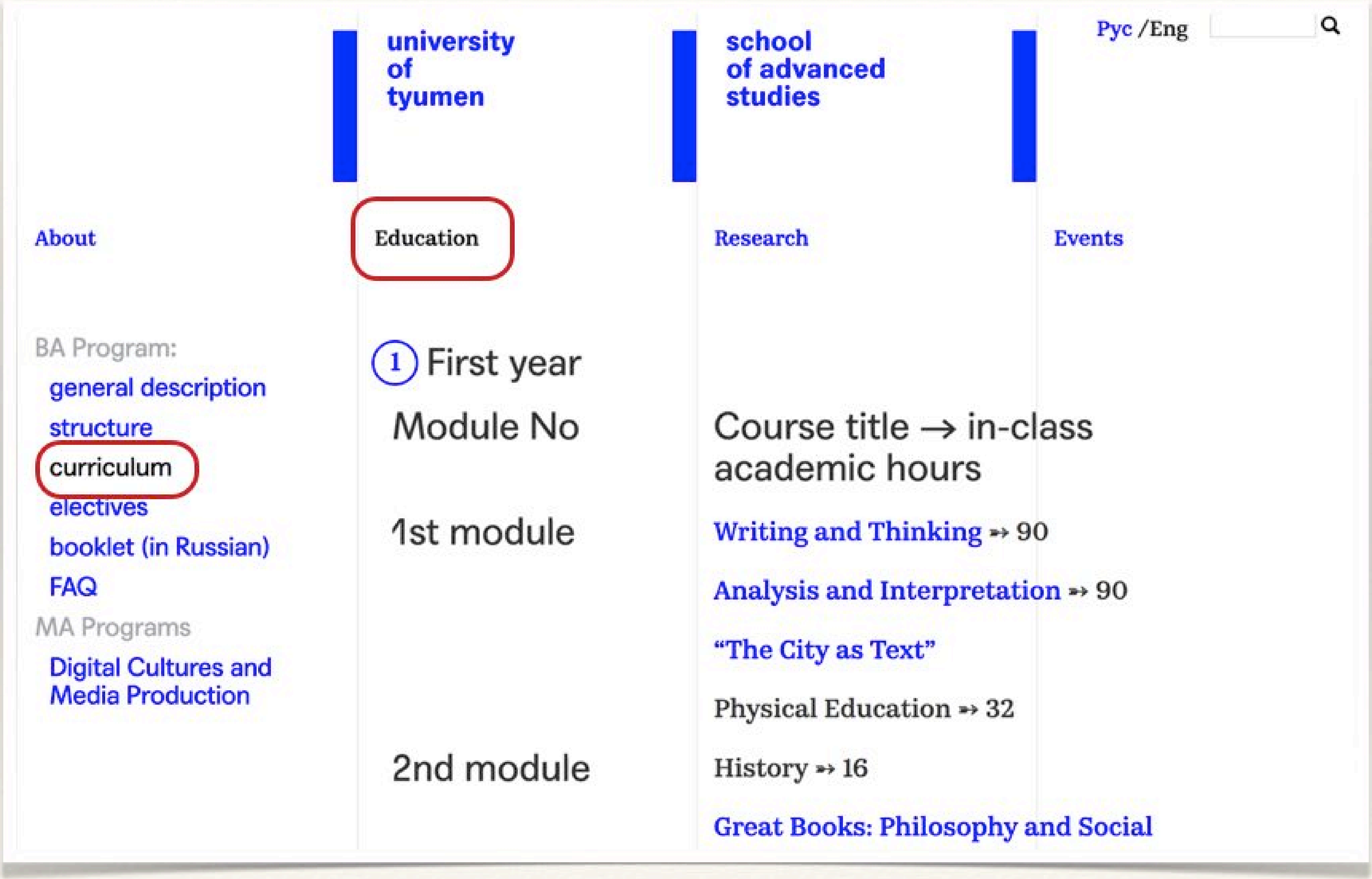
Fabio Grazioso - *17 April 2018*

Course introduction

Course info

- Fabio Grazioso
- email f.grazioso@utmn.ru
- course email sas.it.course@gmail.com
- course web page: https://sas.utmn.ru/en/sas-it-course/
- shared google folder: https://goo.gl/SsXpSr

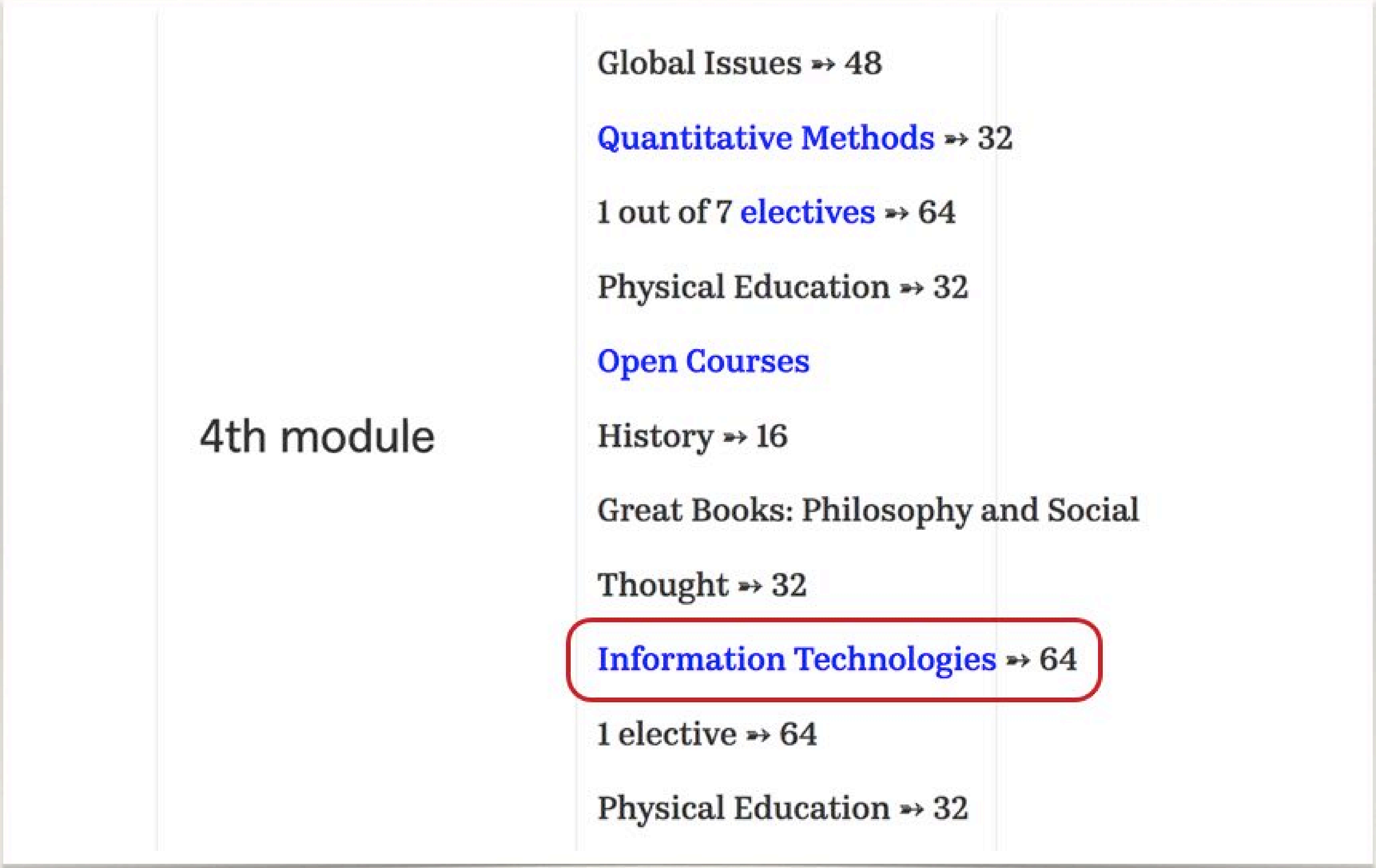
Course web page



① First year

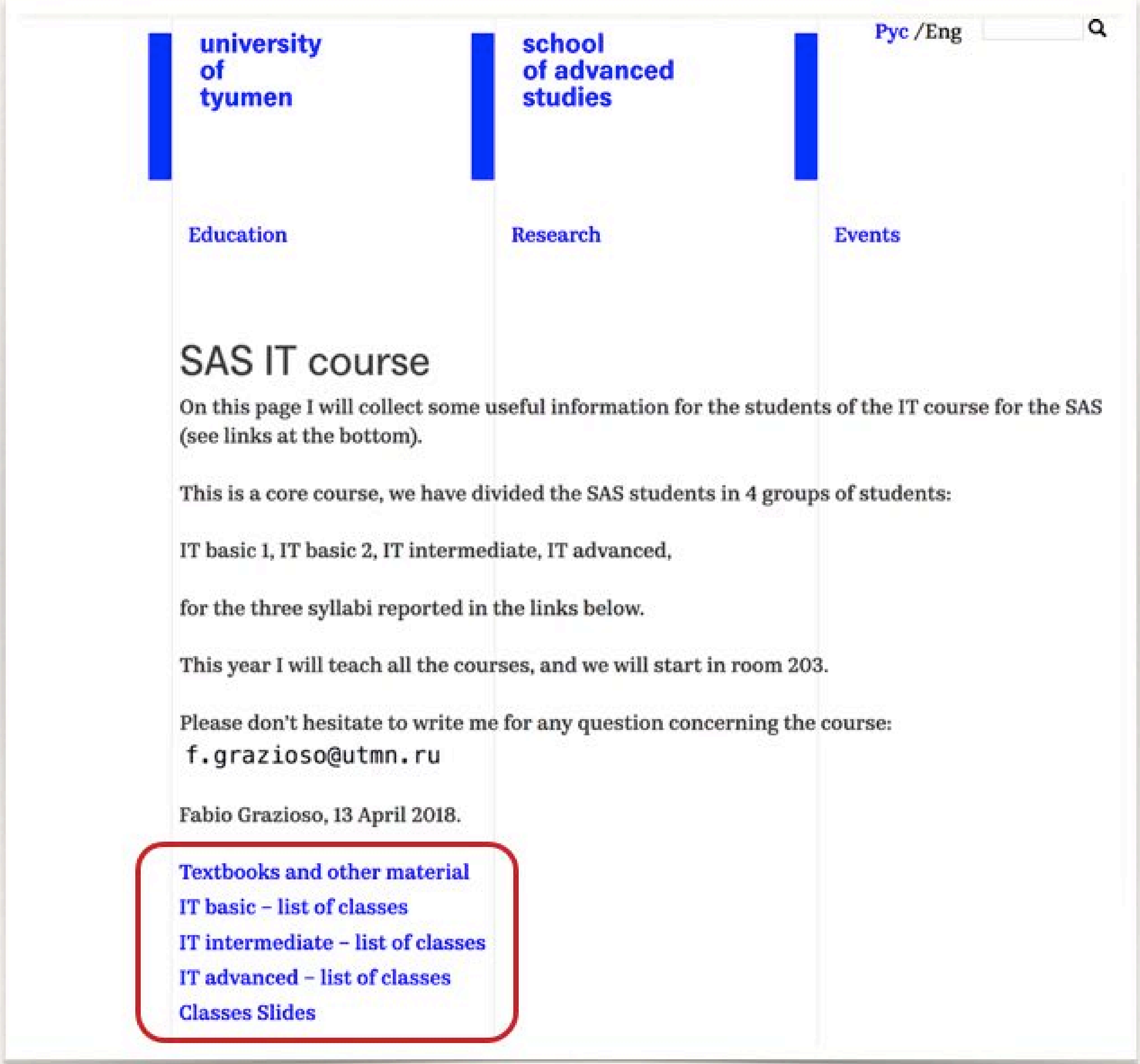
Module No	Course title → in-class academic hours
1st module	Writing and Thinking ↠ 90
	Analysis and Interpretation ↠ 90
	"The City as Text"
	Physical Education ↠ 32
2nd module	History ↠ 16
	Great Books: Philosophy and Social

Course web page



4th module	Global Issues ↠ 48 Quantitative Methods ↠ 32 1 out of 7 electives ↠ 64 Physical Education ↠ 32 Open Courses History ↠ 16 Great Books: Philosophy and Social Thought ↠ 32 Information Technologies ↠ 64 1 elective ↠ 64 Physical Education ↠ 32

Course web page



SAS IT course

On this page I will collect some useful information for the students of the IT course for the SAS (see links at the bottom).

This is a core course, we have divided the SAS students in 4 groups of students:

IT basic 1, IT basic 2, IT intermediate, IT advanced,

for the three syllabi reported in the links below.

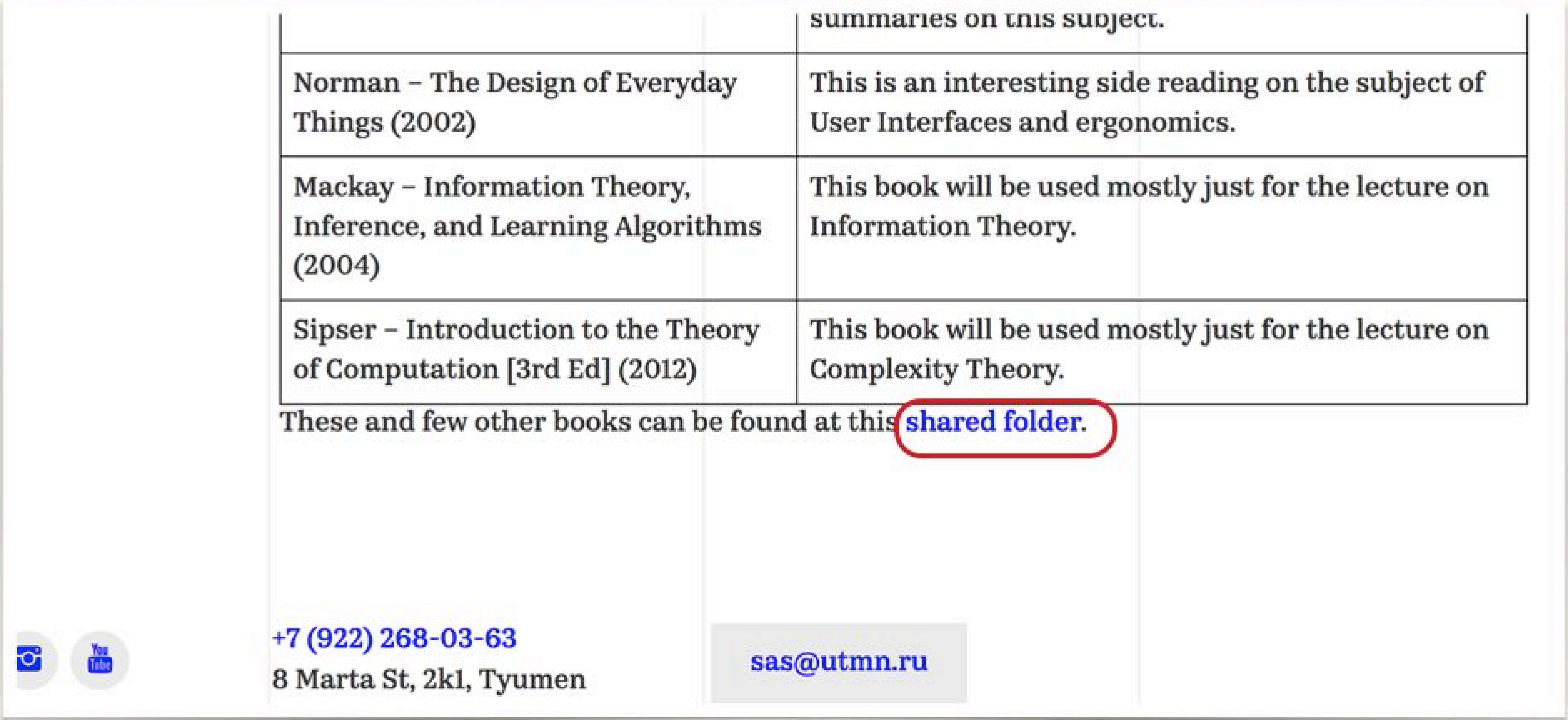
This year I will teach all the courses, and we will start in room 203.

Please don't hesitate to write me for any question concerning the course: `f.grazioso@utmn.ru`

Fabio Grazioso, 13 April 2018.

Textbooks and other material
IT basic – list of classes
IT intermediate – list of classes
IT advanced – list of classes
Classes Slides

Shared folder



	summaries on this subject.
Norman – The Design of Everyday Things (2002)	This is an interesting side reading on the subject of User Interfaces and ergonomics.
Mackay – Information Theory, Inference, and Learning Algorithms (2004)	This book will be used mostly just for the lecture on Information Theory.
Sipser – Introduction to the Theory of Computation [3rd Ed] (2012)	This book will be used mostly just for the lecture on Complexity Theory.

These and few other books can be found at this shared folder.

+7 (922) 268-03-63
8 Marta St, 2k1, Tyumen

sas@utmn.ru

introduction

- bird’s-eye view
- several concepts
- have a general idea of many different things
- have the tools to do an in-depth study in the future
- refer to the books
- you don’t need to read all the books!

Author p##

Todays’ lecture

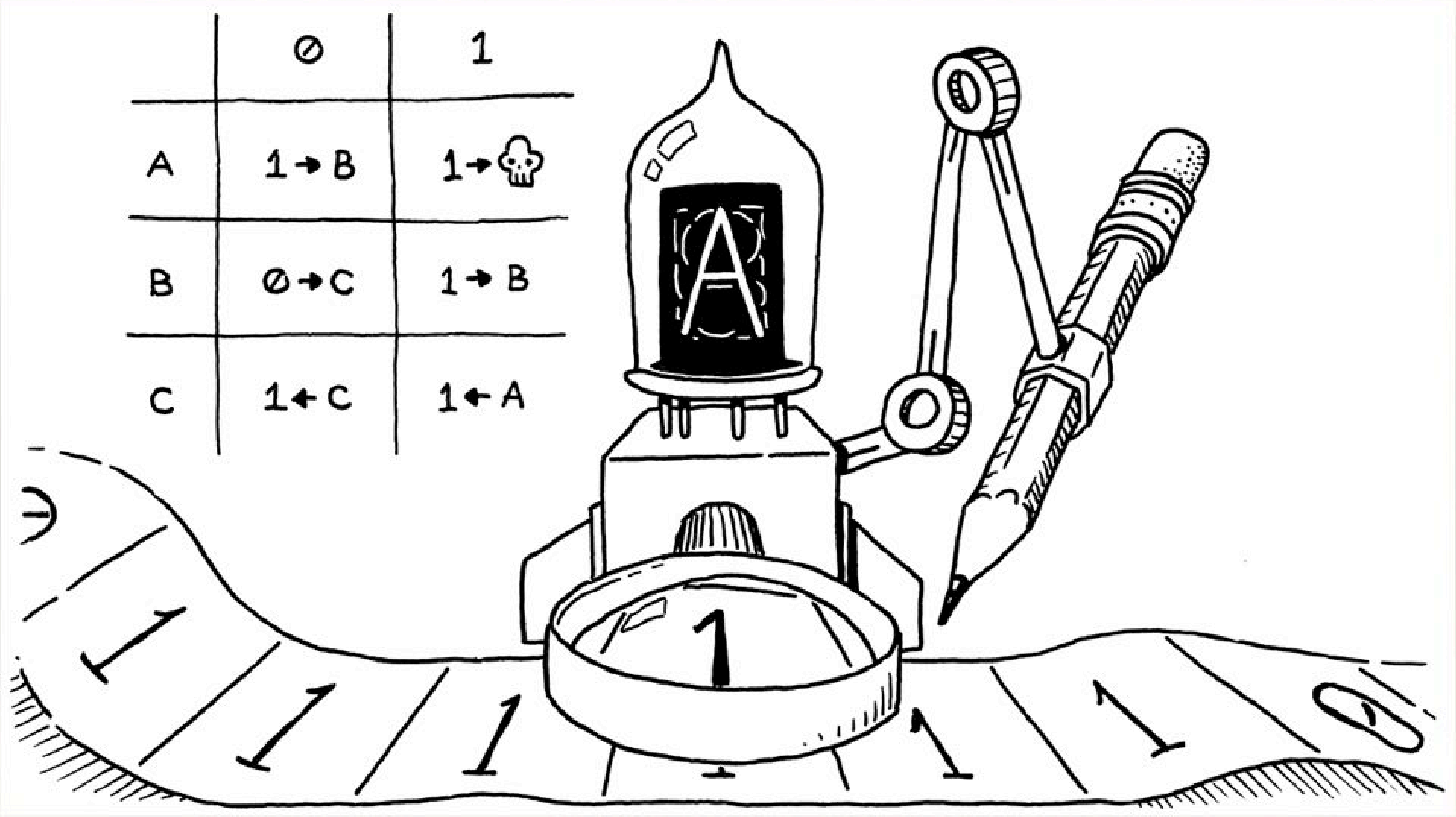
computer problem-solving

summary of the lecture

- Computer architecture (Turing Machine)
- Numbering systems
- Operative Systems and User Interfaces
  - Command Line Interfaces
  - Graphical User Interfaces
- Software interfaces and ergonomics

Computer architecture

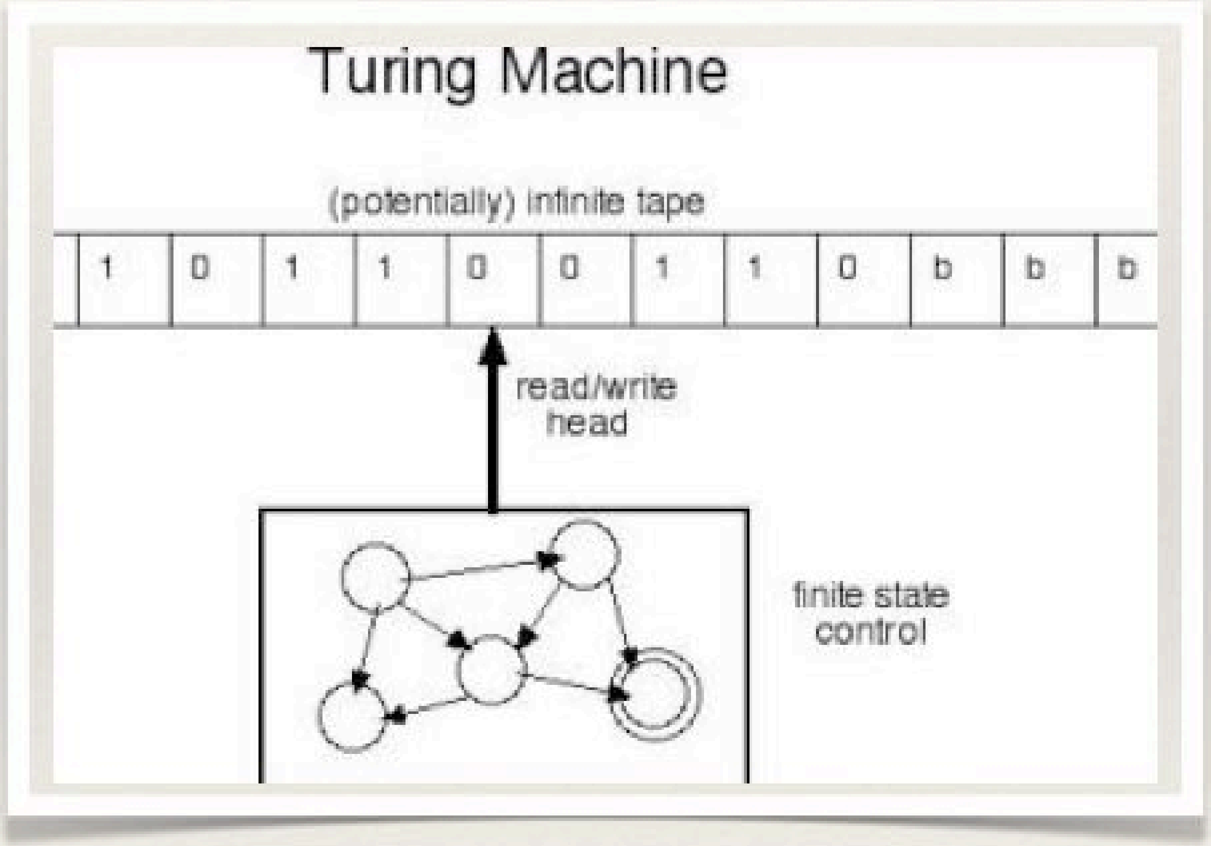
The Turing machine



	Ø	1
A	1→B	1→☠
B	Ø→C	1→B
C	1←C	1←A

The Turing machine

- an infinite tape
  - divided in discrete cells
- a read / write head
- a finite-state control unit

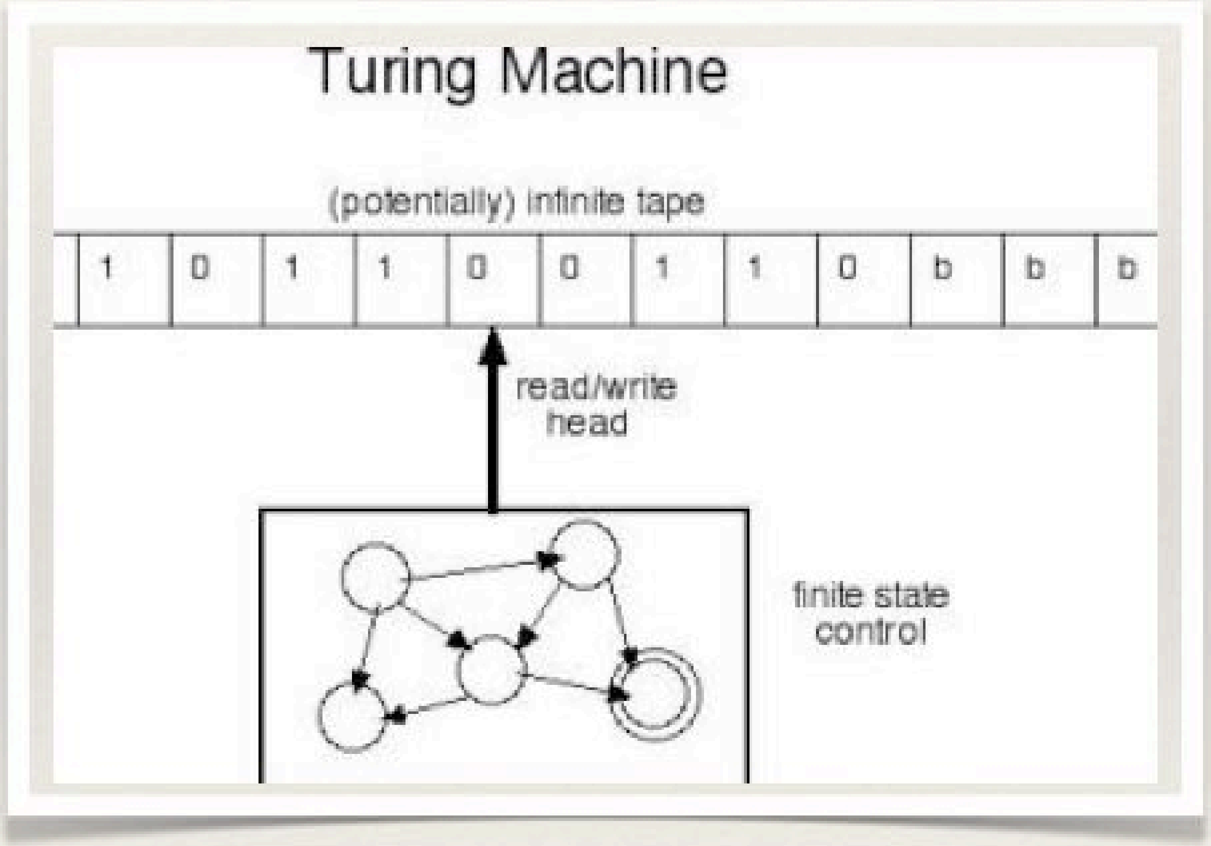


1	0	1	1	0	0	1	1	0	b	b	b

Sipser p165

The Turing machine

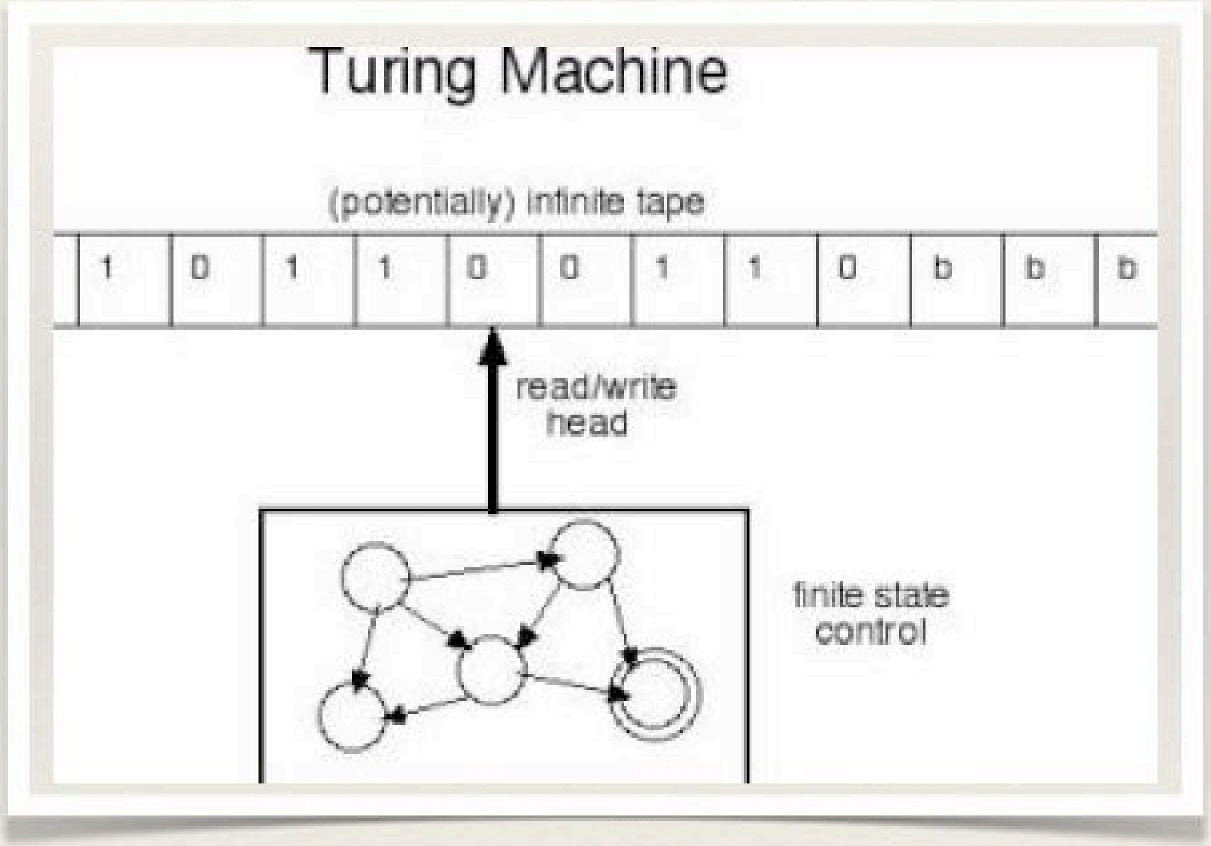
- an infinite tape
  - divided in discrete cells
- a read / write head
- a finite-state control unit



Sipser p165

The Turing machine

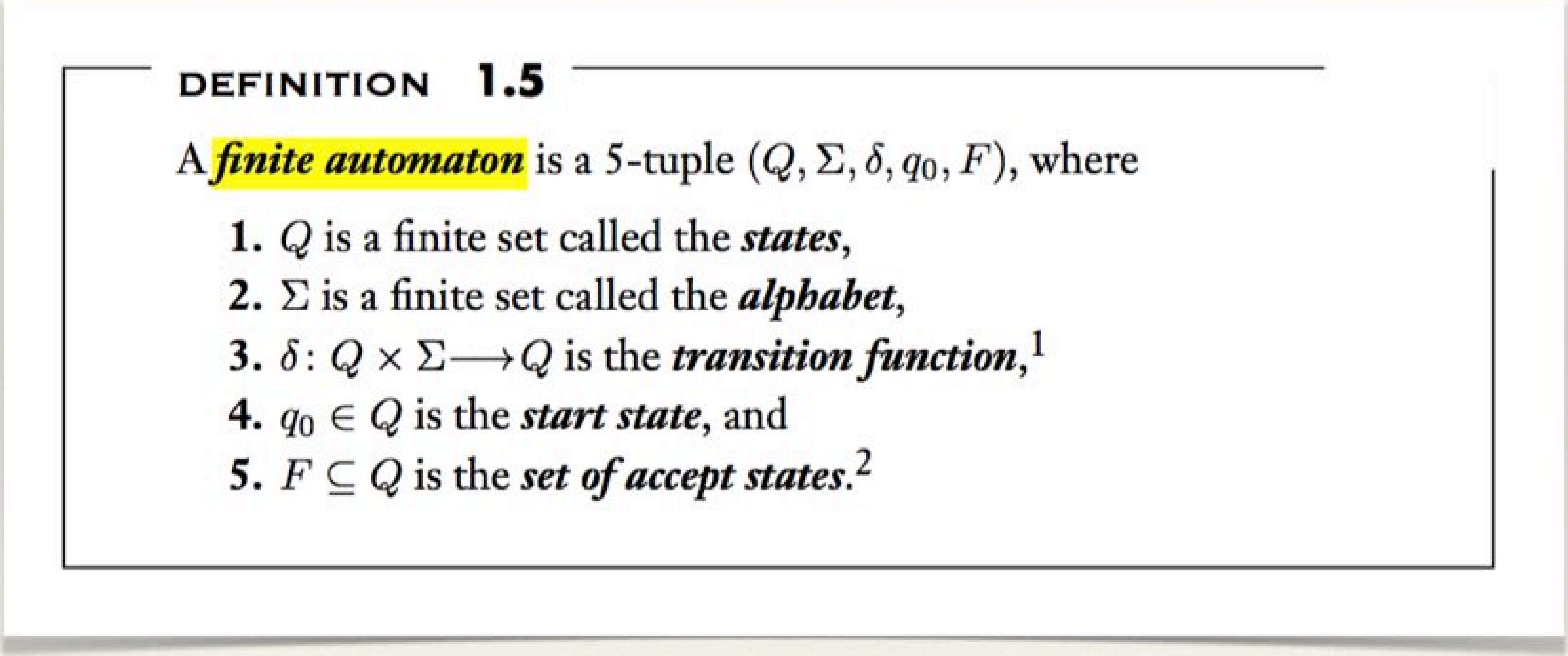
- an infinite tape
  - divided in discrete cells
- a read / write head
- a finite-state control unit



Sipser p165

The Turing machine

- a finite-state control unit



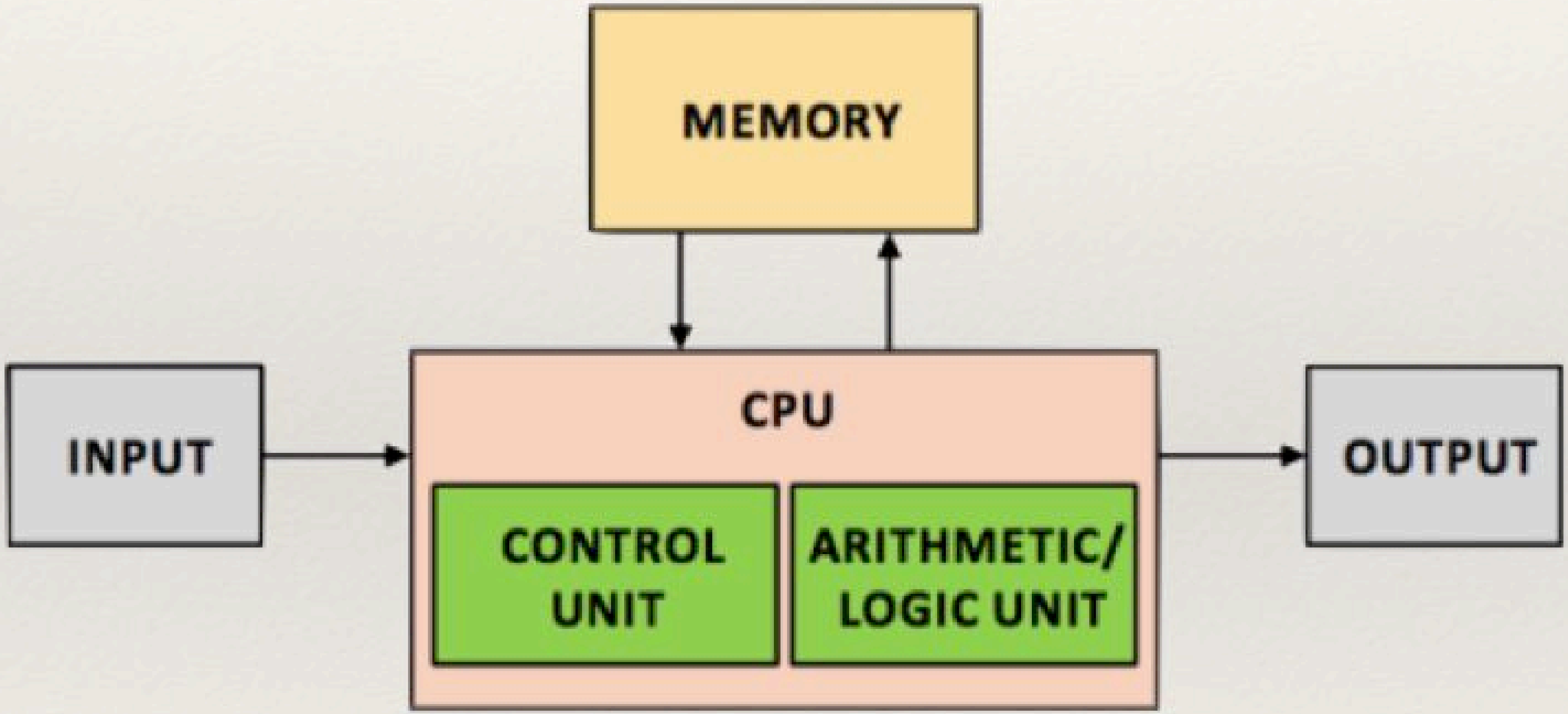
DEFINITION 1.5

A ***finite automaton*** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. Q is a finite set called the ***states***,
2. Σ is a finite set called the ***alphabet***,
3. $\delta \colon Q \times \Sigma \longrightarrow Q$ is the ***transition function***,[1]
4. $q_0 \in Q$ is the ***start state***, and
5. $F \subseteq Q$ is the ***set of accept states***.[2]

Sipser p35

The von Neumann architecture

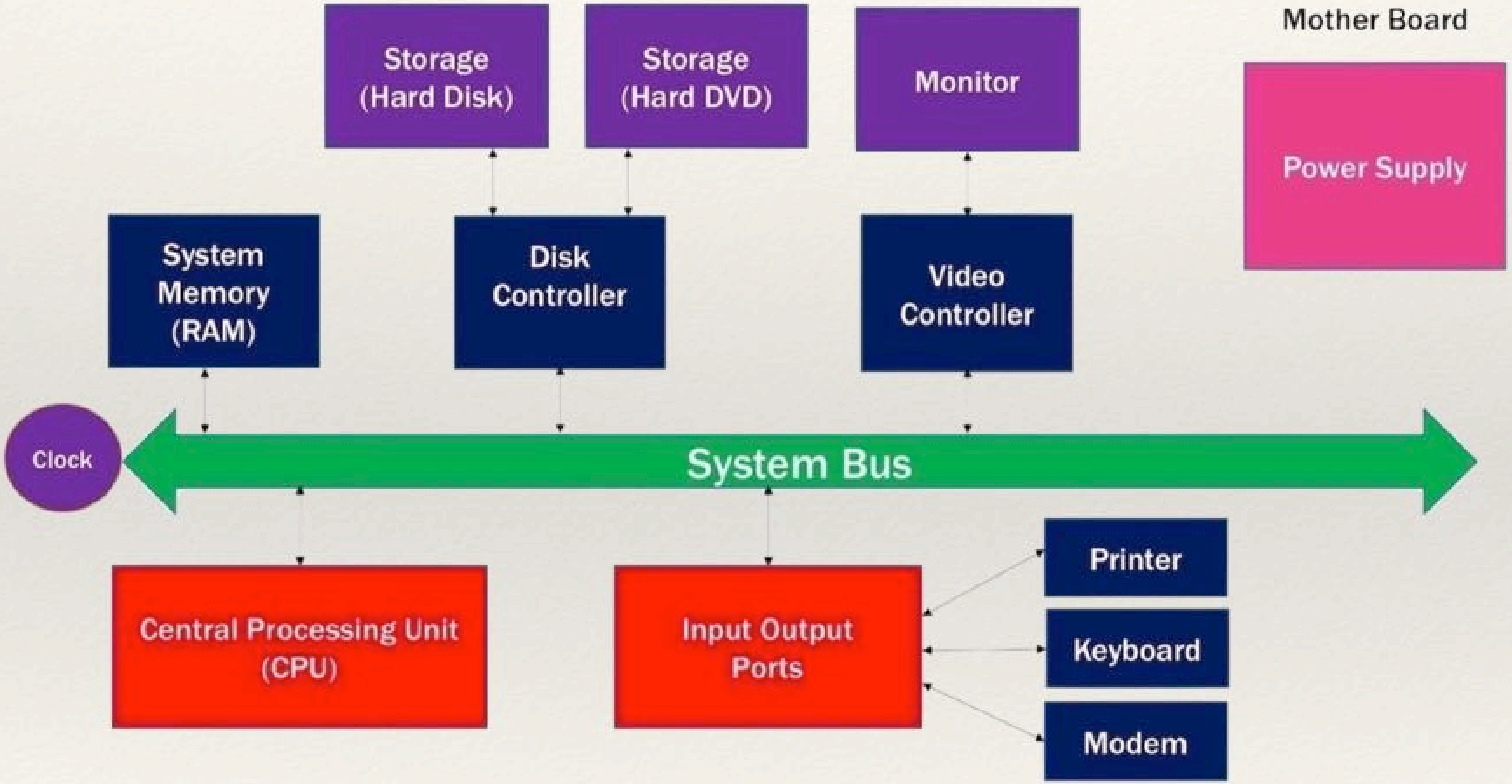


Computer hardware

- System unit
  - motherboard
    - CPU
    - cooling unit for the CPU
    - Possibly extra processors (for instance, for graphics)
    - Memory chips for RAM, ROM
    - Connectors for peripherals (sometimes known as ports)
    - Expansion slots for other peripheral device cards
    - ROM BIOS for booting and basic input and output instructions
    - Power supply connector
  - Disk drives
  - Fan units
  - Power supply
- A monitor
- A keyboard and a pointing device (mouse, track point, track ball)
- Speakers (optional)

Fox p13

Computer hardware



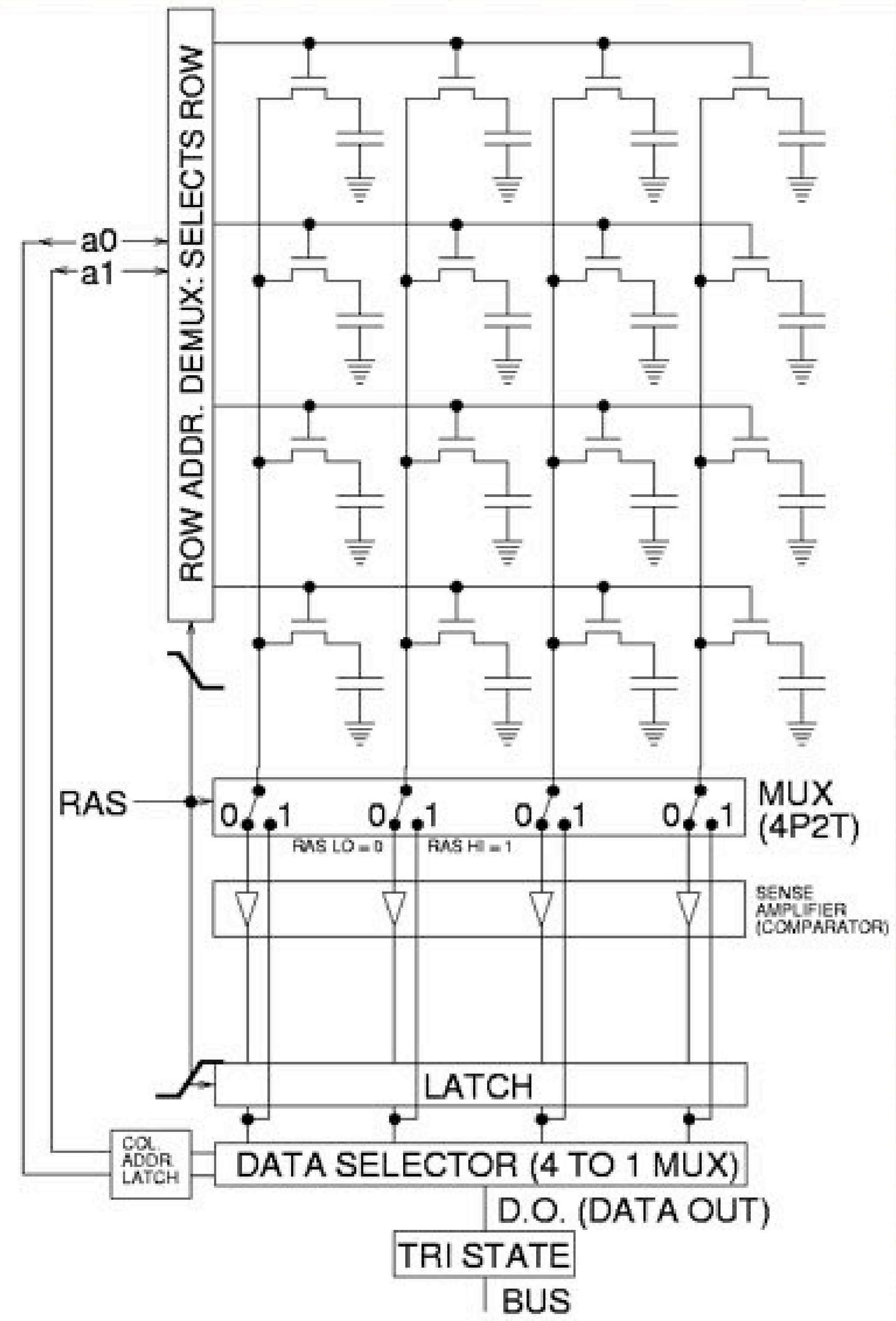
Numbering systems

📖 Fox p57

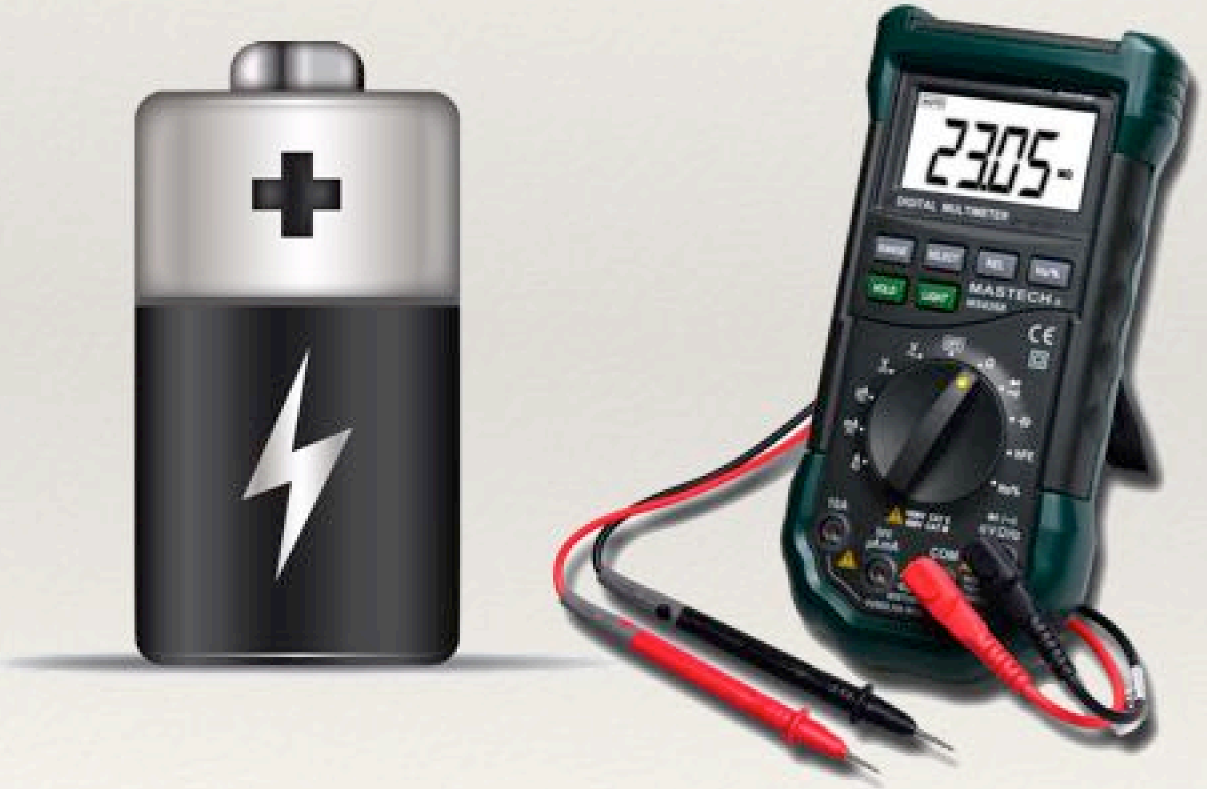
binary numbers

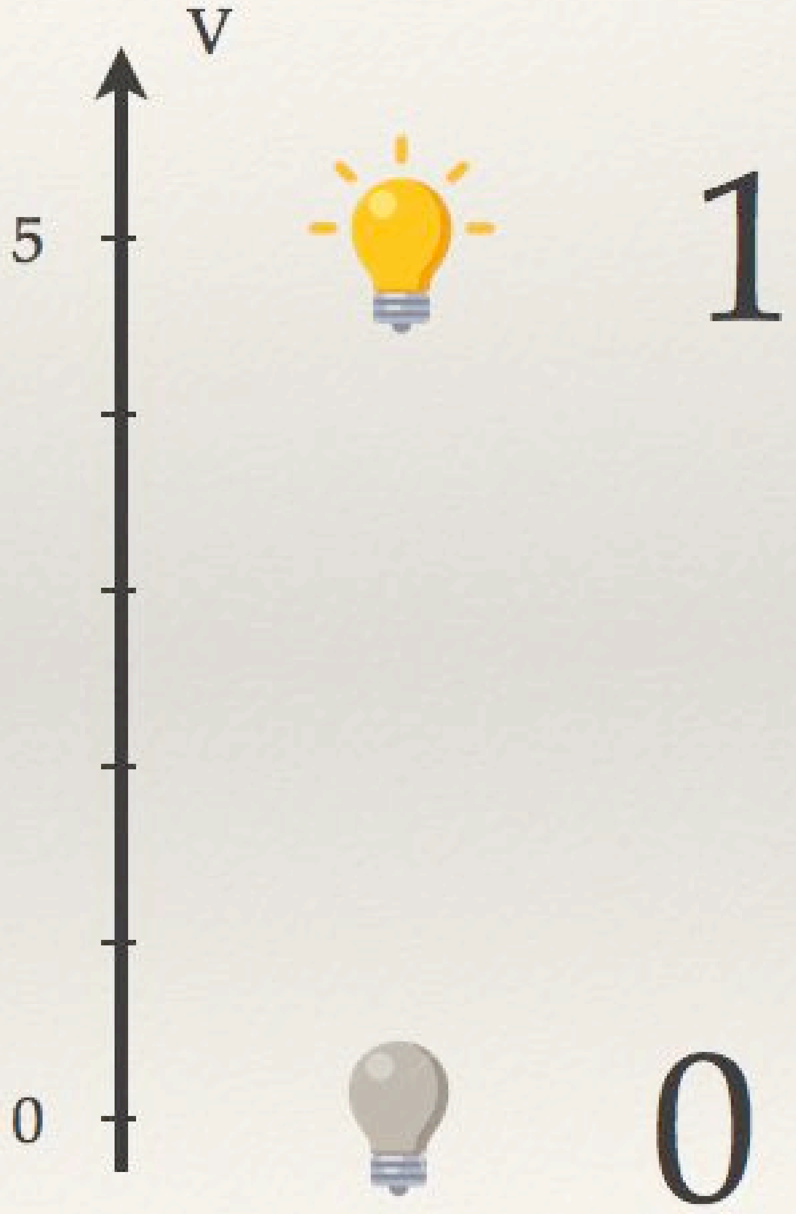
0 0 1 1 1 0 1 0

binary numbers storage

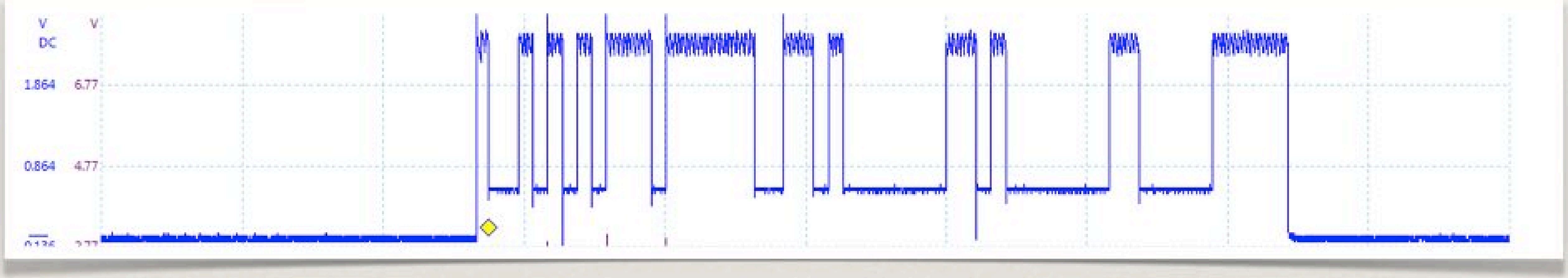


binary numbers storage





binary numbers transmission



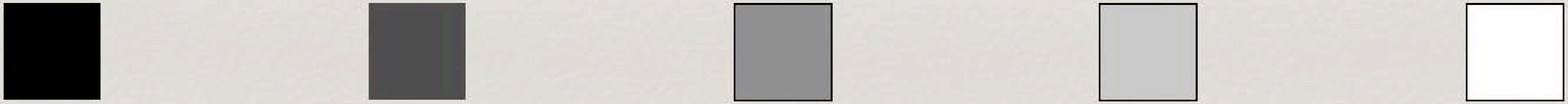
decimal numbers

54729

5 4 7 2 9

decimal numbers

54729



$10000 = 10^4$	$1000 = 10^3$	$100 = 10^2$	$10 = 10^1$	$1 = 10^0$
5	4	7	2	9

binary numbers

$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
■	▩	▒	░	□
1	0	1	1	0

$$= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 16 + 0 + 4 + 2 + 0$$

$$= 22$$

bits and bytes

bit = binary digit

1

byte = 8 bites

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	0	1	1	0

bits and bytes

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
little endian	1	0	1	0	0	1	1	0

	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7
big endian	1	0	1	0	0	1	1	0

floating point - precision

scientific notation

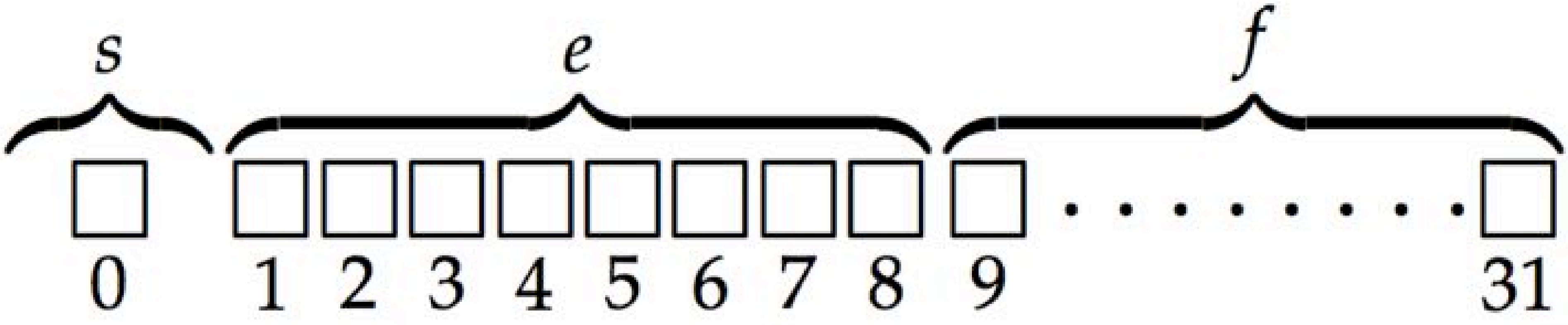
$$100 = 10 \times 10$$

$$100 = 10^2$$

$$500 = 5 \times 100$$

$$500 = 5 \times 10^2$$

floating point



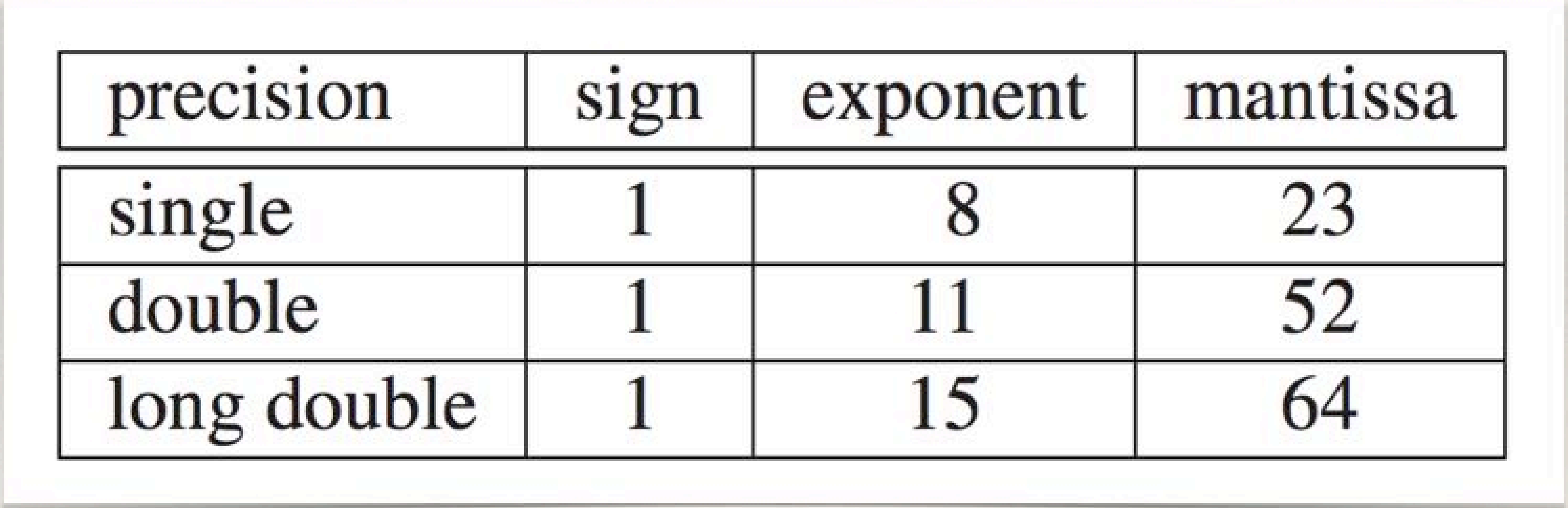
$$\# = (-1)^s \times 2^{e-127} \times 1.\mathrm{f}$$

A floating point number consists of three parts: the **sign** (+ or −), a **mantissa**, which contains the string of significant bits, and an **exponent**. The three parts are stored together in a single computer word.

Sauer p8

floating point

There are three commonly used levels of precision for floating point numbers: single precision, double precision, and extended precision, also known as long-double precision. The number of bits allocated for each floating point number in the three formats is 32,64, and 80, respectively. The bits are divided among the parts as follows:



precision	sign	exponent	mantissa
single	1	8	23
double	1	11	52
long double	1	15	64

Sauer p8

Numerical Analysis

We will study more in details the floating point binaries in a future lecture, when we will study **Numerical Analysis.**

Hexadecimal

hexadecimals

half a byte = 4 bites

$2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$\#\#\#\#$

highest value

$2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$$1111_2 = 15_{10}$$

hexadecimals

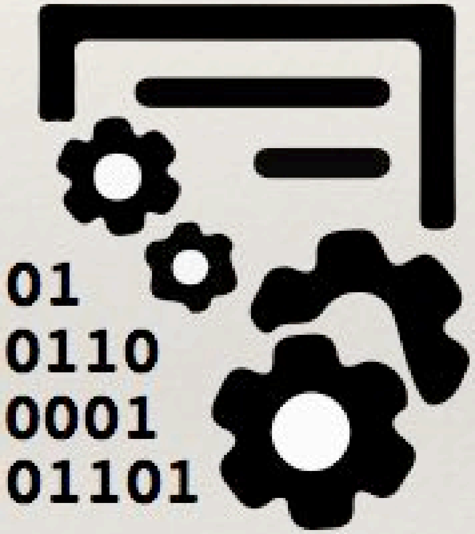
half a byte = 4 bites	2^3 2^2 2^1 2^0 ####
highest value	2^3 2^2 2^1 2^0 $1111_2 = 15_{10}$
hexadecimal	16^3 16^2 16^1 16^0 $\#\#\#\#_{16}$
16 symbols	0123456789ABCDEF

hexadecimals

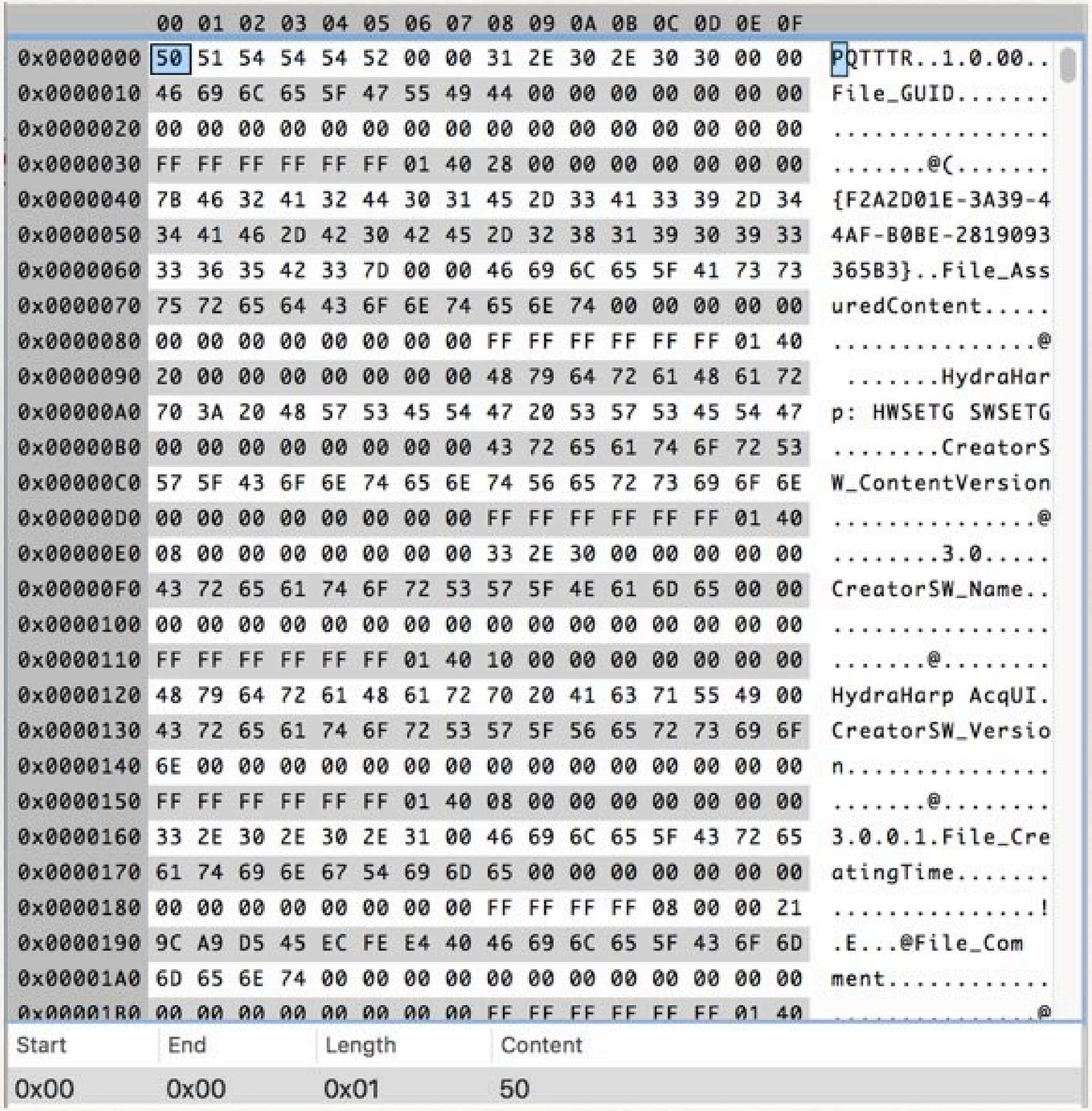
byte = 8 bites 11010110_2

byte = two hex numbers $B5_{16}$

Binary files



Edit (visualize) Binary files

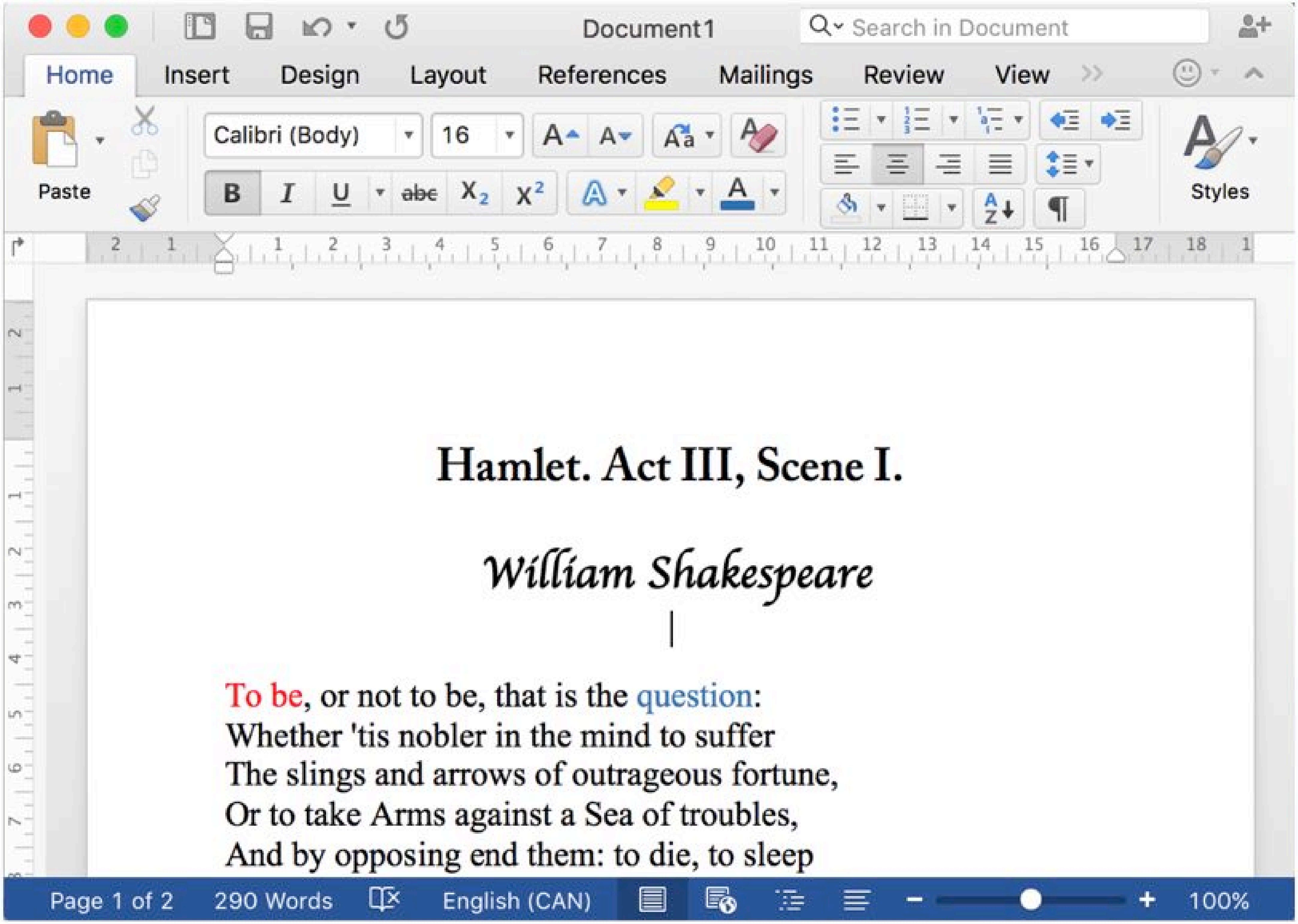


```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0x0000000 50 51 54 54 54 52 00 00 31 2E 30 2E 30 30 00 00  PQTTTR..1.0.00..
0x0000010 46 69 6C 65 5F 47 55 49 44 00 00 00 00 00 00 00  File_GUID.......
0x0000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0000030 FF FF FF FF FF FF 01 40 28 00 00 00 00 00 00 00  .......@(.......
0x0000040 7B 46 32 41 32 44 30 31 45 2D 33 41 33 39 2D 34  {F2A2D01E-3A39-4
0x0000050 34 41 46 2D 42 30 42 45 2D 32 38 31 39 30 39 33  4AF-B0BE-2819093
0x0000060 33 36 35 42 33 7D 00 00 46 69 6C 65 5F 41 73 73  365B3}..File_Ass
0x0000070 75 72 65 64 43 6F 6E 74 65 6E 74 00 00 00 00 00  uredContent.....
0x0000080 00 00 00 00 00 00 00 00 FF FF FF FF FF FF 01 40  ...............@
0x0000090 20 00 00 00 00 00 00 00 48 79 64 72 61 48 61 72   .......HydraHar
0x00000A0 70 3A 20 48 57 53 45 54 47 20 53 57 53 45 54 47  p: HWSETG SWSETG
0x00000B0 00 00 00 00 00 00 00 00 43 72 65 61 74 6F 72 53  ........CreatorS
0x00000C0 57 5F 43 6F 6E 74 65 6E 74 56 65 72 73 69 6F 6E  W_ContentVersion
0x00000D0 00 00 00 00 00 00 00 00 FF FF FF FF FF FF 01 40  ...............@
0x00000E0 08 00 00 00 00 00 00 00 33 2E 30 00 00 00 00 00  ........3.0.....
0x00000F0 43 72 65 61 74 6F 72 53 57 5F 4E 61 6D 65 00 00  CreatorSW_Name..
0x0000100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0000110 FF FF FF FF FF FF 01 40 10 00 00 00 00 00 00 00  .......@........
0x0000120 48 79 64 72 61 48 61 72 70 20 41 63 71 55 49 00  HydraHarp AcqUI.
0x0000130 43 72 65 61 74 6F 72 53 57 5F 56 65 72 73 69 6F  CreatorSW_Versio
0x0000140 6E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  n...............
0x0000150 FF FF FF FF FF FF 01 40 08 00 00 00 00 00 00 00  .......@........
0x0000160 33 2E 30 2E 30 2E 31 00 46 69 6C 65 5F 43 72 65  3.0.0.1.File_Cre
0x0000170 61 74 69 6E 67 54 69 6D 65 00 00 00 00 00 00 00  atingTime.......
0x0000180 00 00 00 00 00 00 00 00 FF FF FF FF 08 00 00 21  ...............!
0x0000190 9C A9 D5 45 EC FE E4 40 46 69 6C 65 5F 43 6F 6D  .E...@File_Com
0x00001A0 6D 65 6E 74 00 00 00 00 00 00 00 00 00 00 00 00  ment............
0x00001B0 00 00 00 00 00 00 00 00 FF FF FF FF FF FF 01 40  ...............@
```

Start	End	Length	Content
0x00	0x00	0x01	50

text encodings

text files

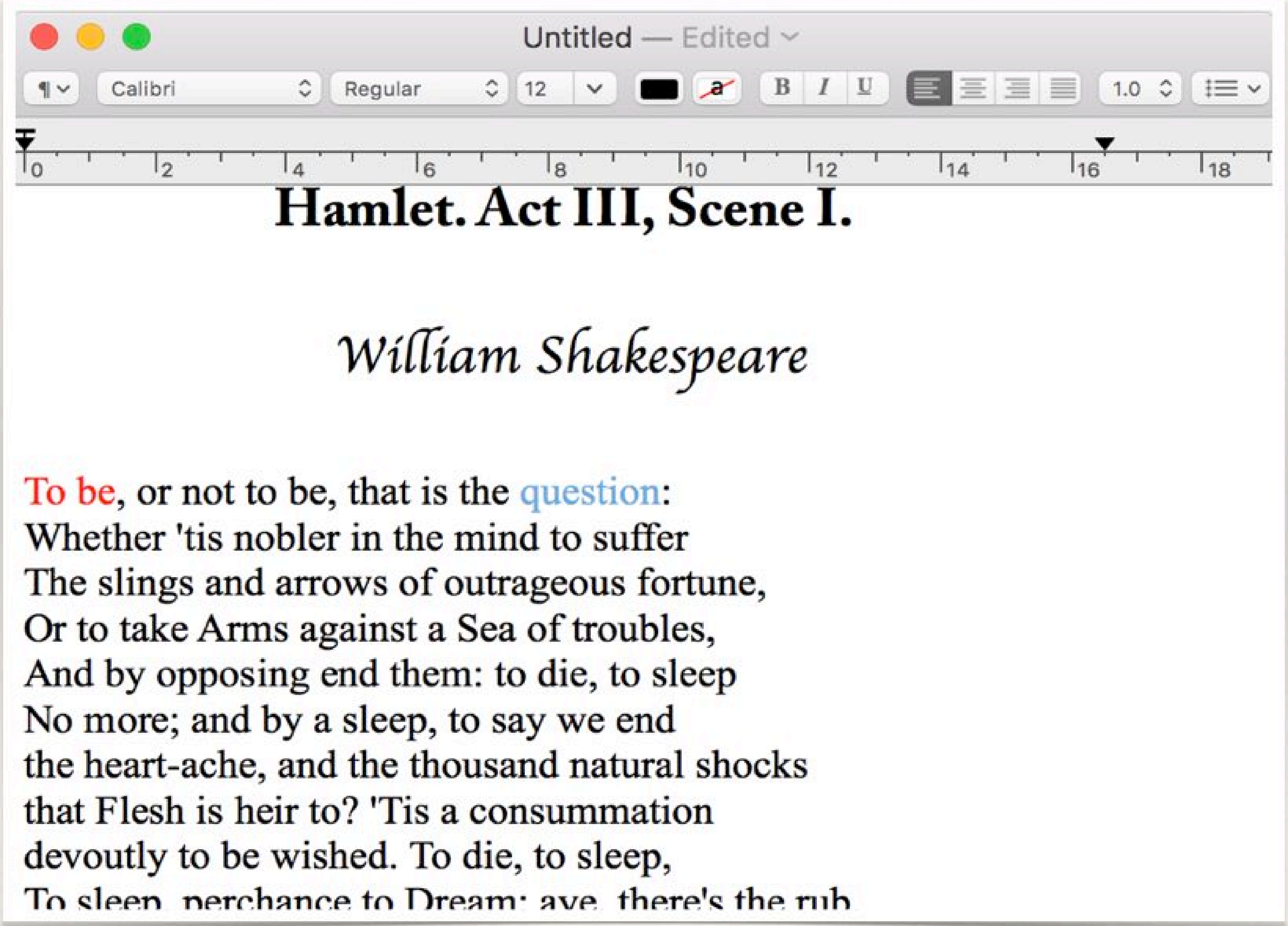


Hamlet. Act III, Scene I.

William Shakespeare

To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep

text files

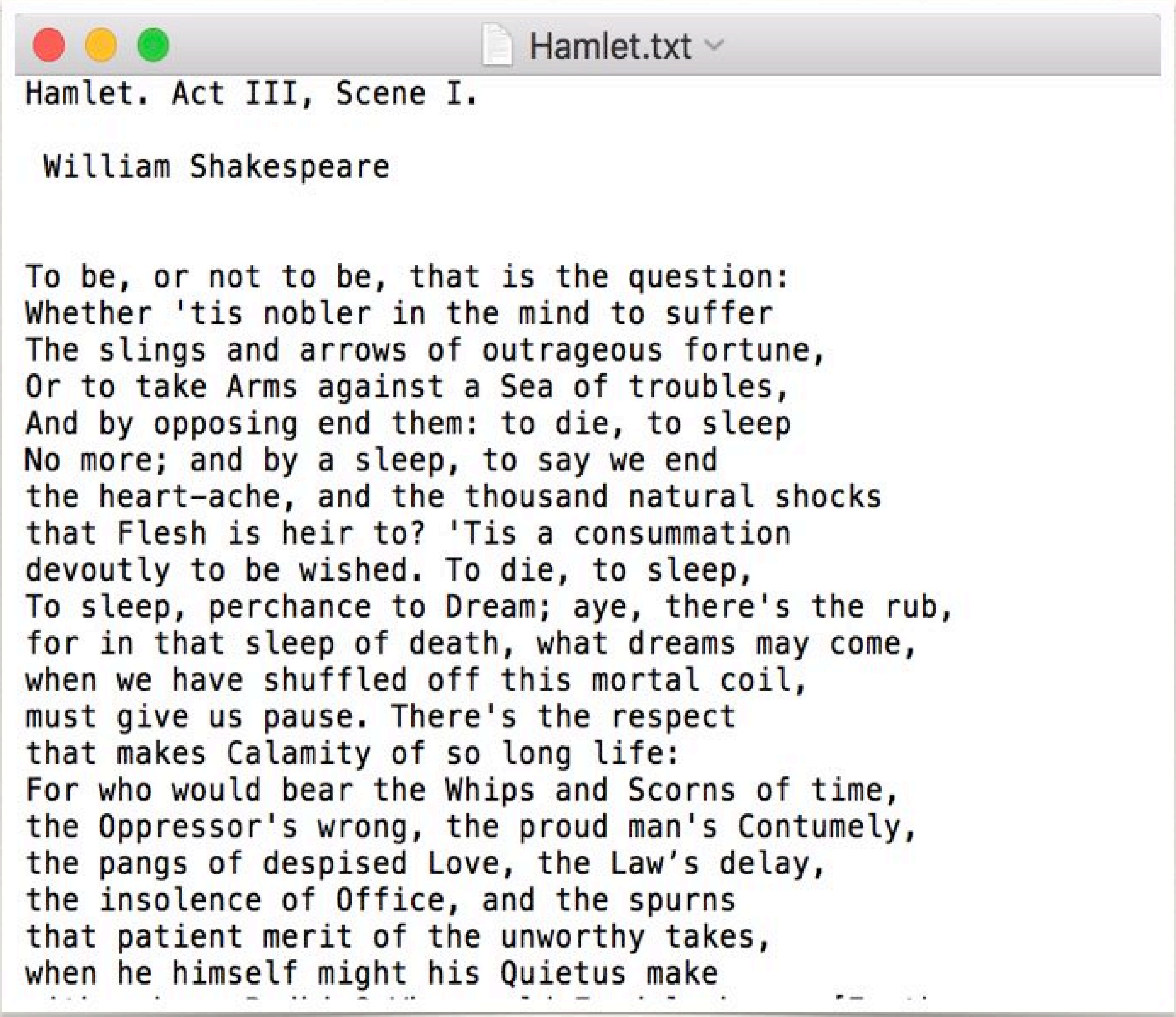


Hamlet. Act III, Scene I.

William Shakespeare

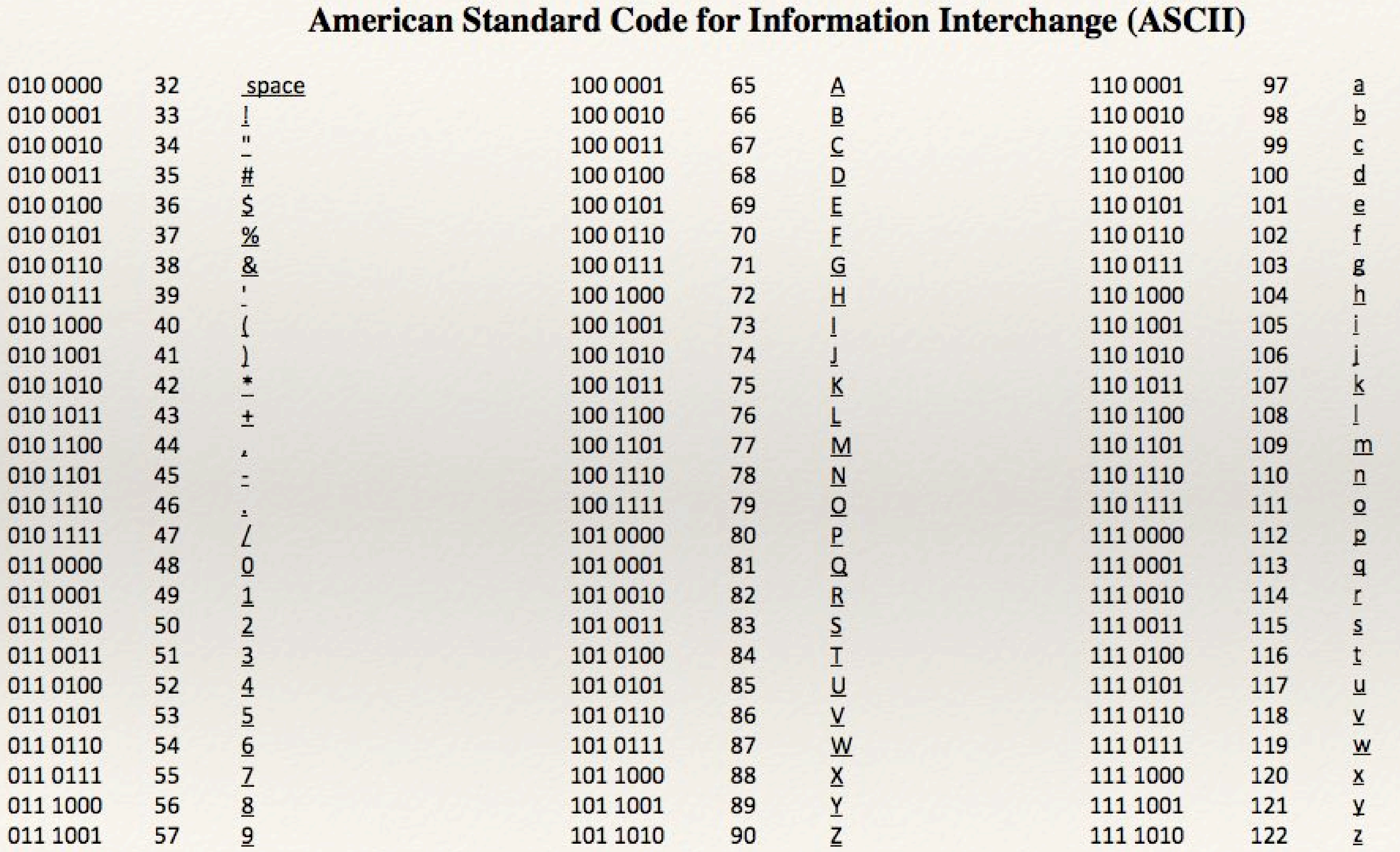
To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
No more; and by a sleep, to say we end
the heart-ache, and the thousand natural shocks
that Flesh is heir to? 'Tis a consummation
devoutly to be wished. To die, to sleep,
To sleep, perchance to Dream; aye, there's the rub

text files



```
Hamlet. Act III, Scene I.

 William Shakespeare


To be, or not to be, that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take Arms against a Sea of troubles,
And by opposing end them: to die, to sleep
No more; and by a sleep, to say we end
the heart-ache, and the thousand natural shocks
that Flesh is heir to? 'Tis a consummation
devoutly to be wished. To die, to sleep,
To sleep, perchance to Dream; aye, there's the rub,
for in that sleep of death, what dreams may come,
when we have shuffled off this mortal coil,
must give us pause. There's the respect
that makes Calamity of so long life:
For who would bear the Whips and Scorns of time,
the Oppressor's wrong, the proud man's Contumely,
the pangs of despised Love, the Law’s delay,
the insolence of Office, and the spurns
that patient merit of the unworthy takes,
when he himself might his Quietus make
```

ascii characters



American Standard Code for Information Interchange (ASCII)

010 0000	32	space	100 0001	65	A	110 0001	97	a
010 0001	33	!	100 0010	66	B	110 0010	98	b
010 0010	34	"	100 0011	67	C	110 0011	99	c
010 0011	35	#	100 0100	68	D	110 0100	100	d
010 0100	36	$	100 0101	69	E	110 0101	101	e
010 0101	37	%	100 0110	70	F	110 0110	102	f
010 0110	38	&	100 0111	71	G	110 0111	103	g
010 0111	39	'	100 1000	72	H	110 1000	104	h
010 1000	40	(	100 1001	73	I	110 1001	105	i
010 1001	41	)	100 1010	74	J	110 1010	106	j
010 1010	42	*	100 1011	75	K	110 1011	107	k
010 1011	43	+	100 1100	76	L	110 1100	108	l
010 1100	44	,	100 1101	77	M	110 1101	109	m
010 1101	45	-	100 1110	78	N	110 1110	110	n
010 1110	46	.	100 1111	79	O	110 1111	111	o
010 1111	47	/	101 0000	80	P	111 0000	112	p
011 0000	48	0	101 0001	81	Q	111 0001	113	q
011 0001	49	1	101 0010	82	R	111 0010	114	r
011 0010	50	2	101 0011	83	S	111 0011	115	s
011 0011	51	3	101 0100	84	T	111 0100	116	t
011 0100	52	4	101 0101	85	U	111 0101	117	u
011 0101	53	5	101 0110	86	V	111 0110	118	v
011 0110	54	6	101 0111	87	W	111 0111	119	w
011 0111	55	7	101 1000	88	X	111 1000	120	x
011 1000	56	8	101 1001	89	Y	111 1001	121	y
011 1001	57	9	101 1010	90	Z	111 1010	122	z

ascii encoding

- The best known and most widely used character encoding standard is the **American Standard Code for Information Interchange** (ASCII).
- The first version of ASCII was published in **1964** as a standard way of representing textual data in computer memory and sending it over communication links between computers.
- ASCII is based on a **seven-bit byte.** Each byte represented a character, and characters were represented by assigning them to individual binary numbers.

Gillam p7

ascii encoding

what is the highest value that we can write with 7 binary digits?

$$\begin{array}{ccccccc} 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

$$\begin{array}{cccccccc} 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$2^7 = 128$$

Gillam p7

ascii encoding

- Perhaps the main deficiency in ASCII comes from the "A" in its name: **American**. ASCII is an American standard, and was designed for the storage and transmission of English text. 95 characters are sufficient for representing English text, barely, but that's it. On early teletype machines, ASCII could also be used to represent the accented letters found in many European languages, but this capability disappeared in the transition from teletypes to **CRT terminals**.

CRT = Cathode Ray Tube

Gillam p7

Unicode

- Unicode is the latest of several attempts to solve this Tower of Babel problem by creating a universal character encoding.
- Its main way of doing this is to increase the size of the possible encoding space by **increasing the number of bits used to encode each character**.
- Most other character encodings are based upon an **eight-bit byte**, which provides enough space to encode a maximum of **256 characters** (in practice, most encodings reserve some of these values for control signals and encode fewer than 256 characters).

Gillam p7

Unicode

- Unicode uses a 16-bit word to encode characters, allowing up to 65,536 characters to be encoded. 65,000 characters, with careful management, is enough to allow encoding of the vast majority of characters in the vast majority of written languages in use today.
- The current version of Unicode, version 3.2, actually encodes 95,156 different characters—it actually does use a scheme to represent the less-common characters using two 16-bit units, but with 50,212 characters actually encoded using only a single unit, you rarely encounter the two-unit characters. In fact, these 50,212 characters include all of the characters representable with all of the other character encoding methods that are in reasonably widespread use.

📖 Gillam p7

UTF-8

- UTF-8 is a variable width character encoding capable of encoding all 1,112,064 valid code points in **Unicode** using one to four 8-bit **bytes**.
- The name is derived from Unicode Transformation Format – 8-bit.

Wikipedia

cyrillic

0410	А	0430	а
0411	Б	0431	б
0412	В	0432	в
0413	Г	0433	г
0414	Д	0434	д
0415	Е	0435	е
0416	Ж	0436	ж
0417	З	0437	з
0418	И	0438	и
0419	Й	0439	й
041A	К	043A	к
041B	Л	043B	л
041C	М	043C	м
041D	Н	043D	н
041E	О	043E	о
041F	П	043F	п

0420	Р	0440	р
0421	С	0441	с
0422	Т	0442	т
0423	У	0443	у
0424	Ф	0444	ф
0425	Х	0445	х
0426	Ц	0446	ц
0427	Ч	0447	ч
0428	Ш	0448	ш
0429	Щ	0449	щ
042A	Ъ	044A	ъ
042B	Ы	044B	ы
042C	Ь	044C	ь
042D	Э	044D	э
042E	Ю	044E	ю
042F	Я	044F	я

Operative Systems

Operative System

- An operating system acts as an **intermediary** between the **user** of a computer and the **computer hardware**.
- The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner.

Tanenbaum p4

UI and GUI

- The Operative System (OS) is the software that lets the user interact with the hardware of a computer.
- How do we communicate with the OS?
- We need an **interface**, something between the user and the OS.

CLI

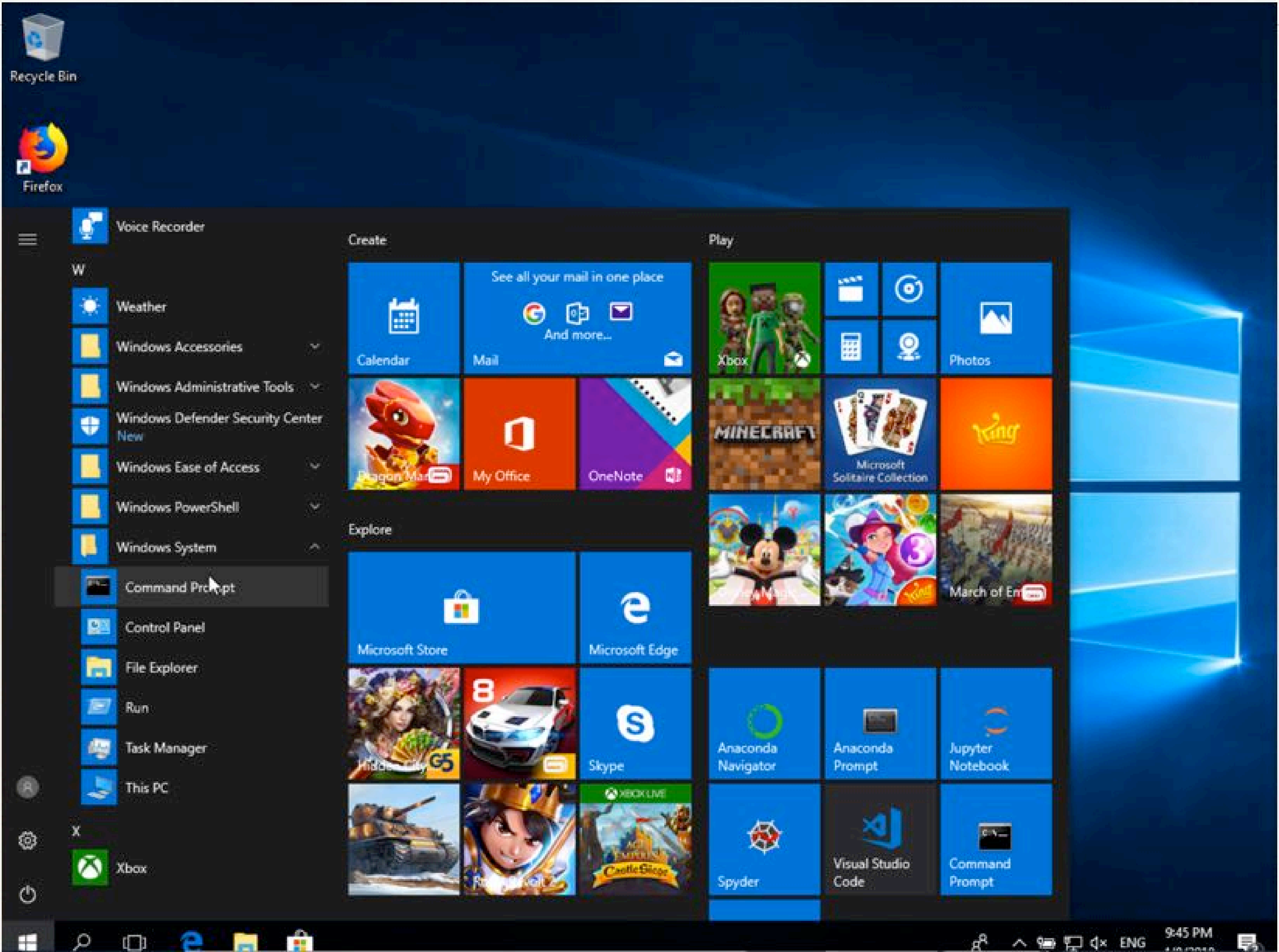
VT100 & VT220 terminals

command-line interface (CLI)

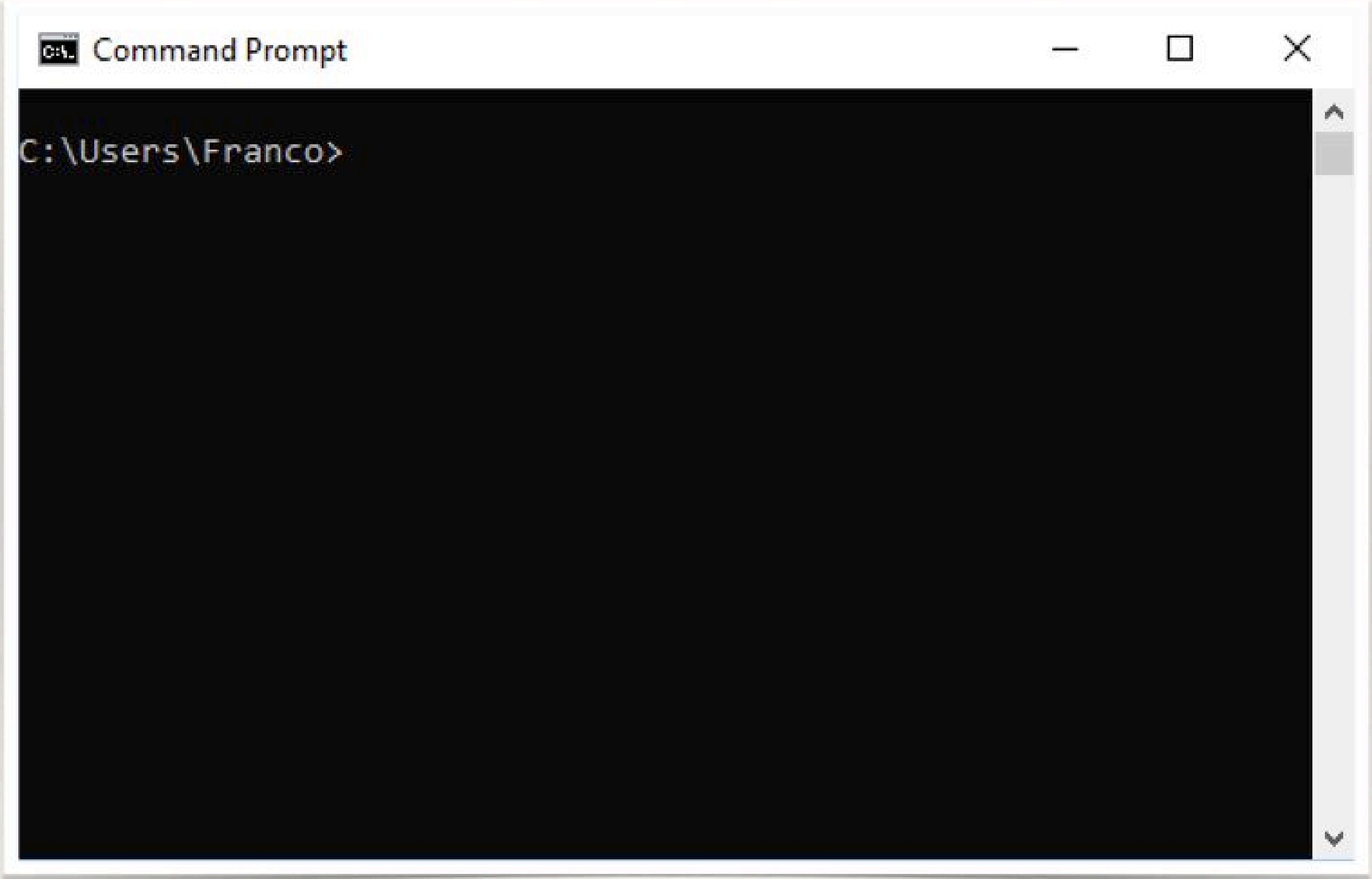
- command-line interface (CLI) has been the typical UI.
- Some well known OS based on CLI were:
    - VAX/VMS (Digital)
    - MS-DOS (Microsoft)
    - ProDOS (Apple)

Windows

Microsoft Windows

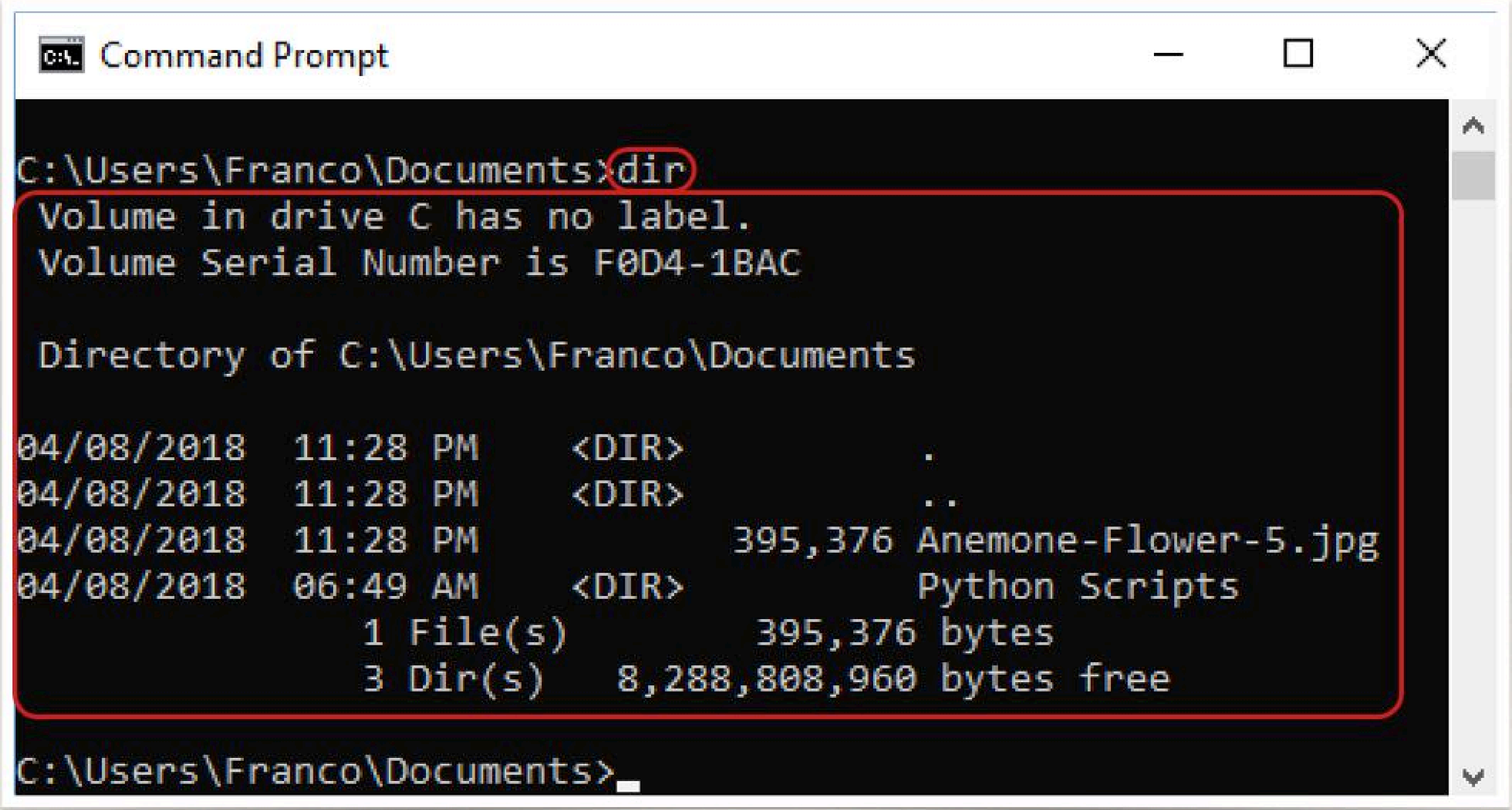


Windows Console



```
C:\Users\Franco>
```

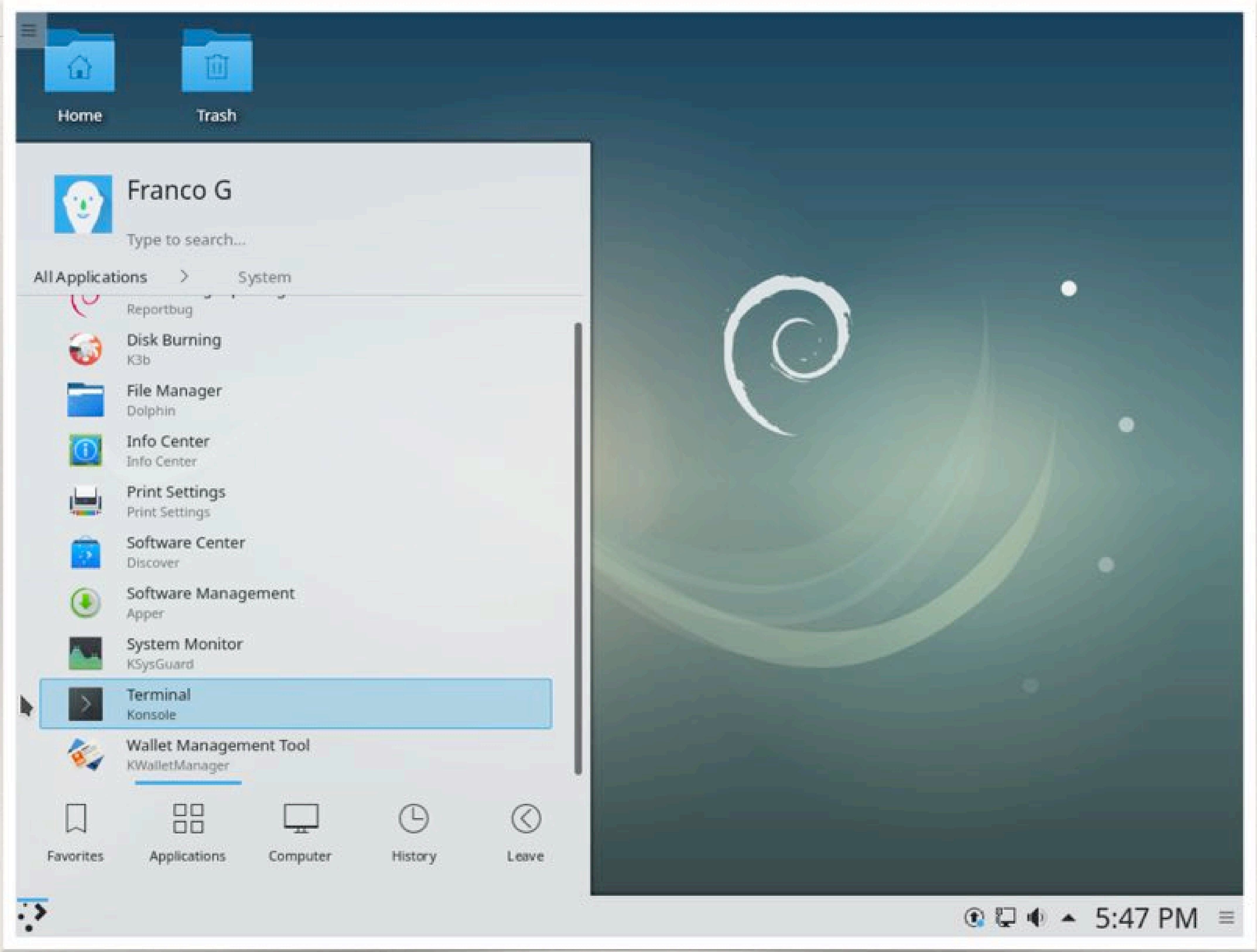
Windows Console



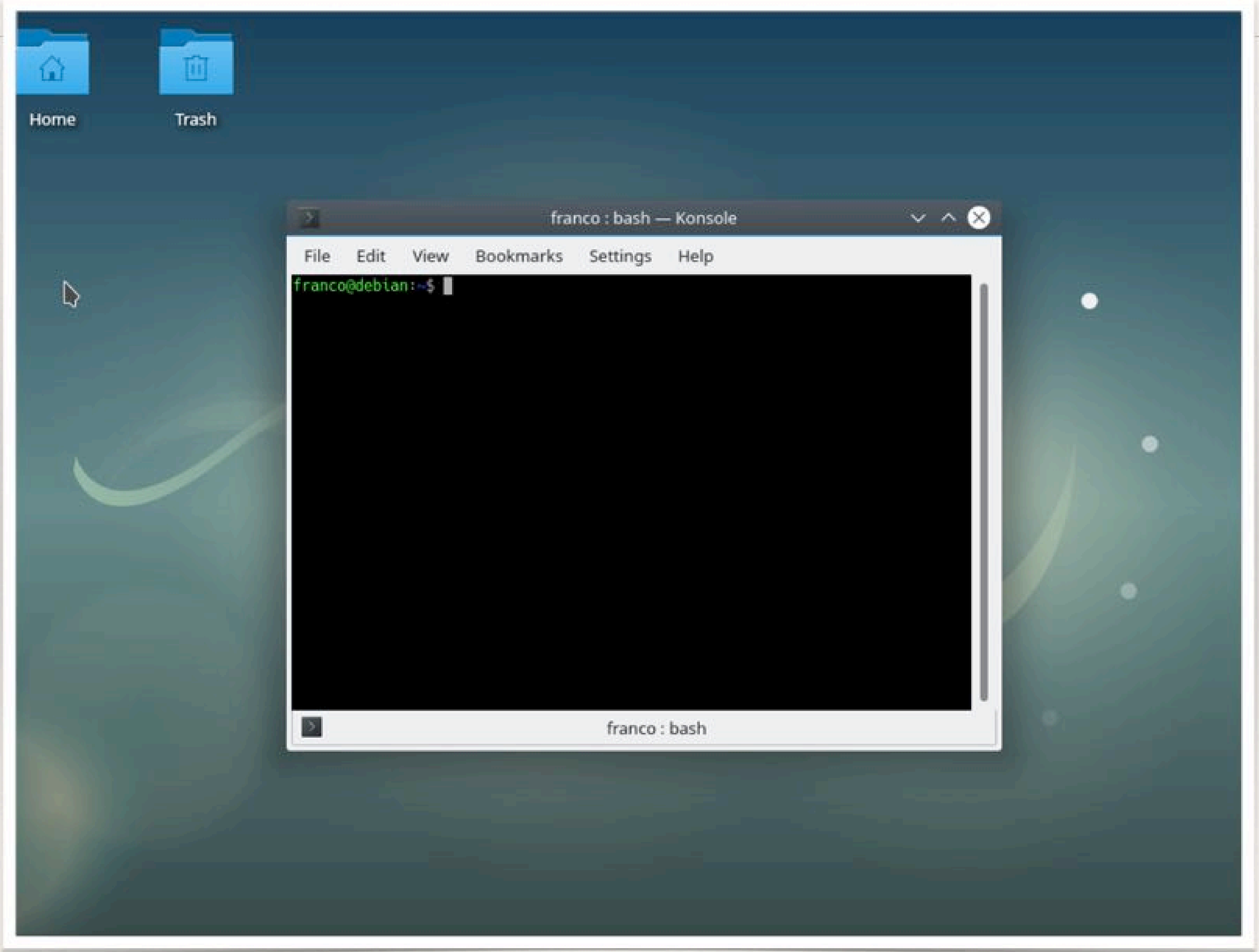
```
C:\Users\Franco\Documents>dir
 Volume in drive C has no label.
 Volume Serial Number is F0D4-1BAC

 Directory of C:\Users\Franco\Documents

04/08/2018  11:28 PM    <DIR>          .
04/08/2018  11:28 PM    <DIR>          ..
04/08/2018  11:28 PM           395,376 Anemone-Flower-5.jpg
04/08/2018  06:49 AM    <DIR>          Python Scripts
               1 File(s)        395,376 bytes
               3 Dir(s)   8,288,808,960 bytes free

C:\Users\Franco\Documents>_
```

Unix/Linux

Debian with KDE

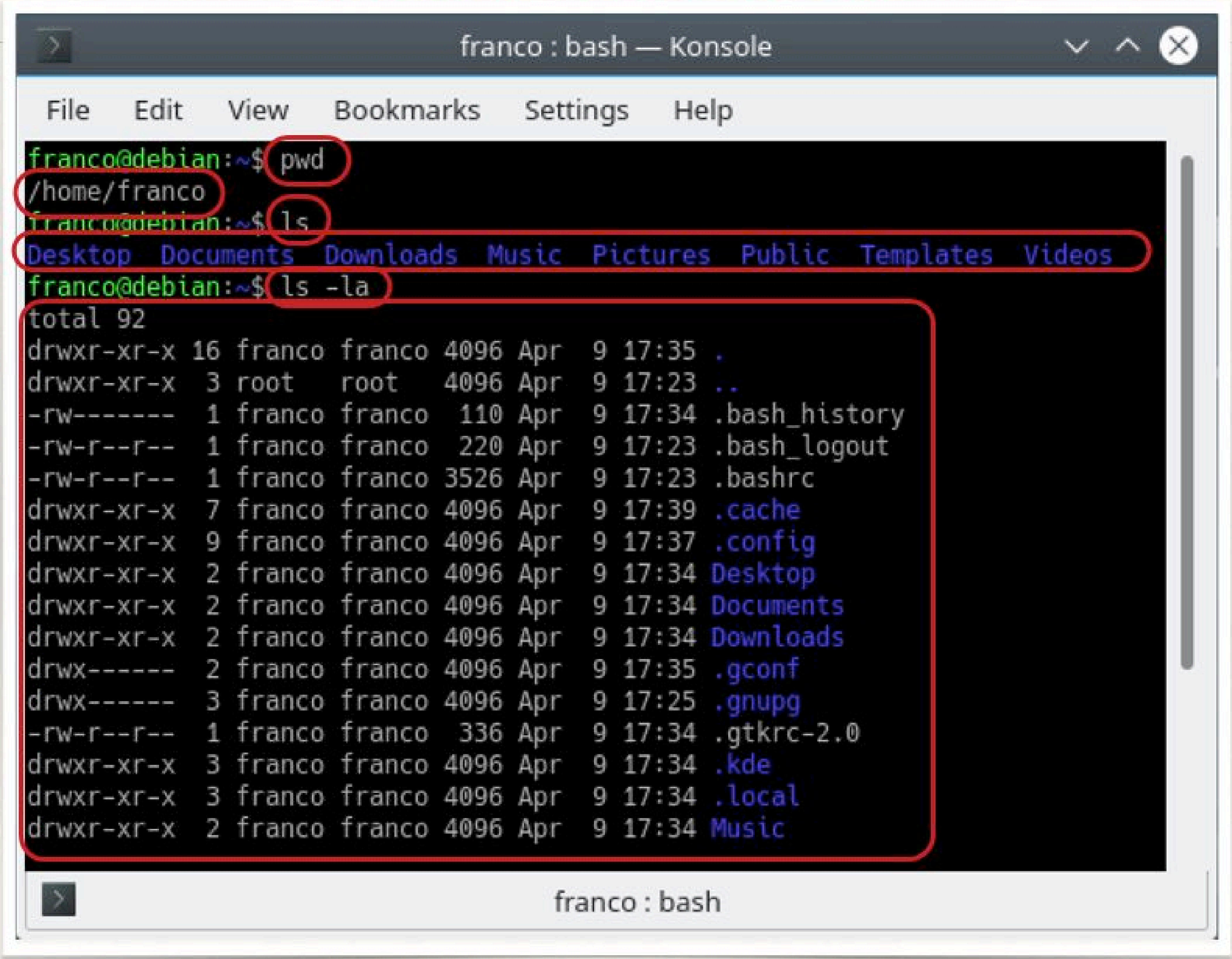


Debian with KDE



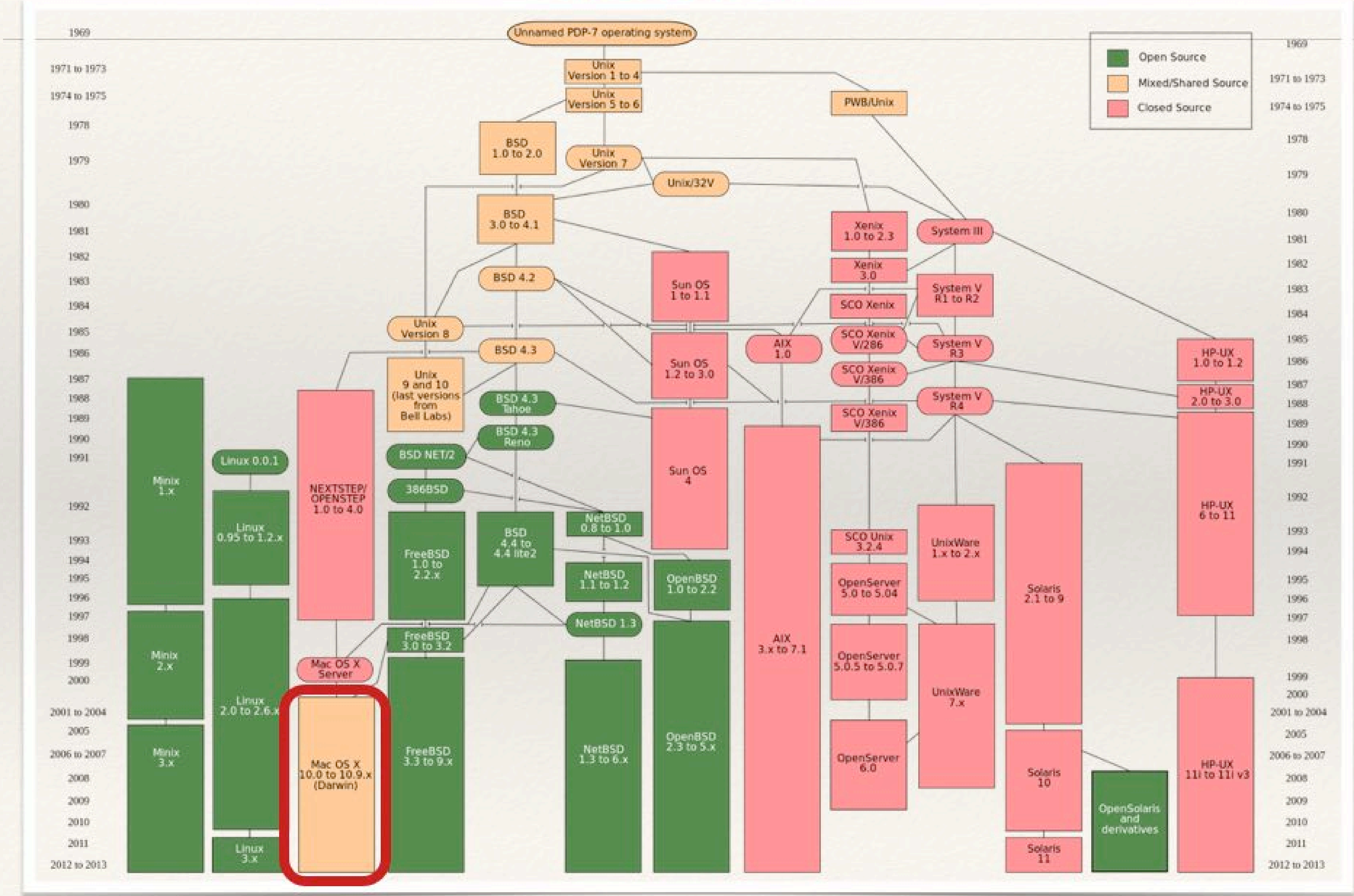
```
franco@debian:~$
```

linux Console

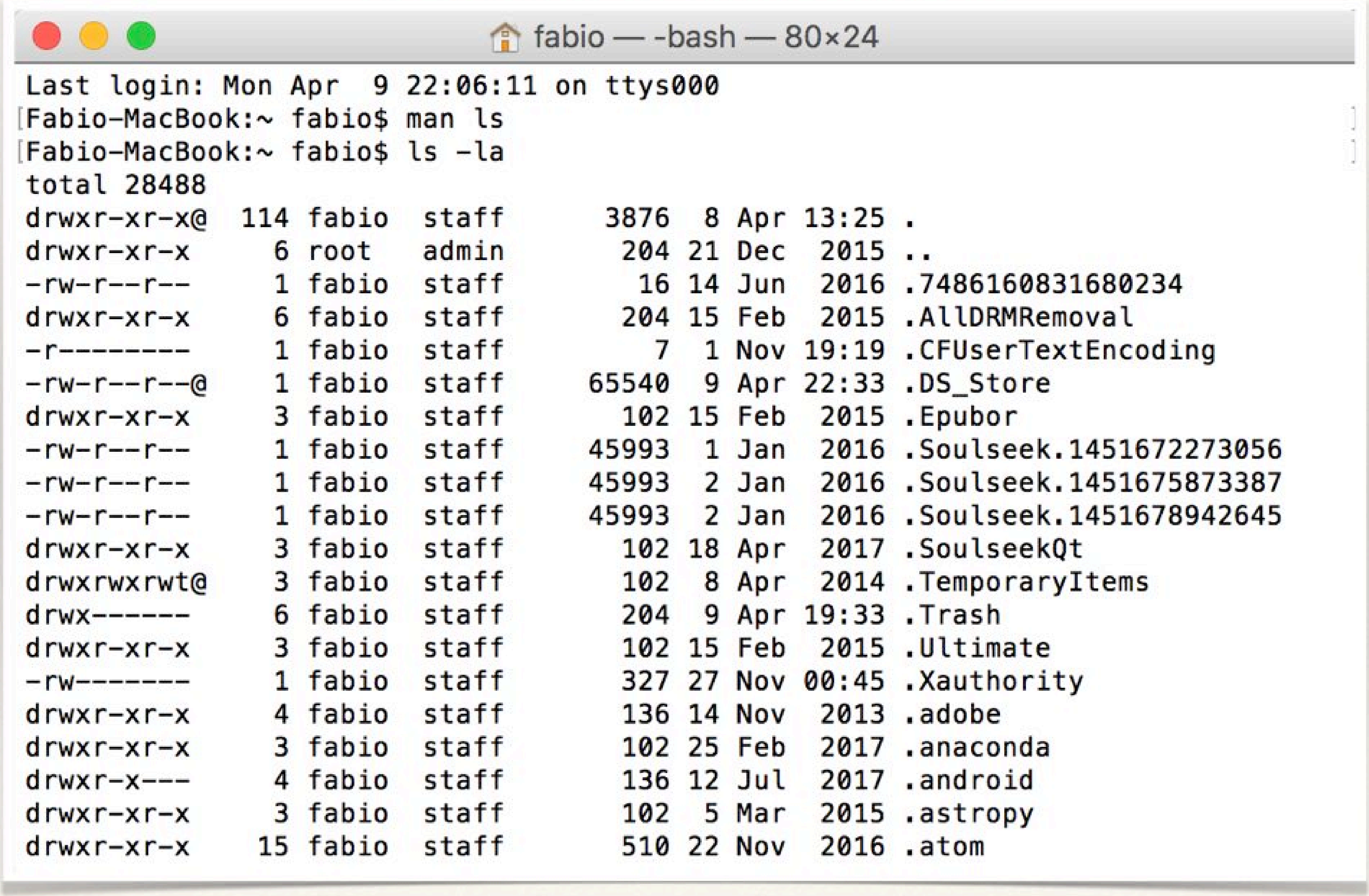


```
franco@debian:~$ pwd
/home/franco
franco@debian:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
franco@debian:~$ ls -la
total 92
drwxr-xr-x 16 franco franco 4096 Apr  9 17:35 .
drwxr-xr-x  3 root   root   4096 Apr  9 17:23 ..
-rw-------  1 franco franco  110 Apr  9 17:34 .bash_history
-rw-r--r--  1 franco franco  220 Apr  9 17:23 .bash_logout
-rw-r--r--  1 franco franco 3526 Apr  9 17:23 .bashrc
drwxr-xr-x  7 franco franco 4096 Apr  9 17:39 .cache
drwxr-xr-x  9 franco franco 4096 Apr  9 17:37 .config
drwxr-xr-x  2 franco franco 4096 Apr  9 17:34 Desktop
drwxr-xr-x  2 franco franco 4096 Apr  9 17:34 Documents
drwxr-xr-x  2 franco franco 4096 Apr  9 17:34 Downloads
drwx------  2 franco franco 4096 Apr  9 17:35 .gconf
drwx------  3 franco franco 4096 Apr  9 17:25 .gnupg
-rw-r--r--  1 franco franco  336 Apr  9 17:34 .gtkrc-2.0
drwxr-xr-x  3 franco franco 4096 Apr  9 17:34 .kde
drwxr-xr-x  3 franco franco 4096 Apr  9 17:34 .local
drwxr-xr-x  2 franco franco 4096 Apr  9 17:34 Music
```

MacOS

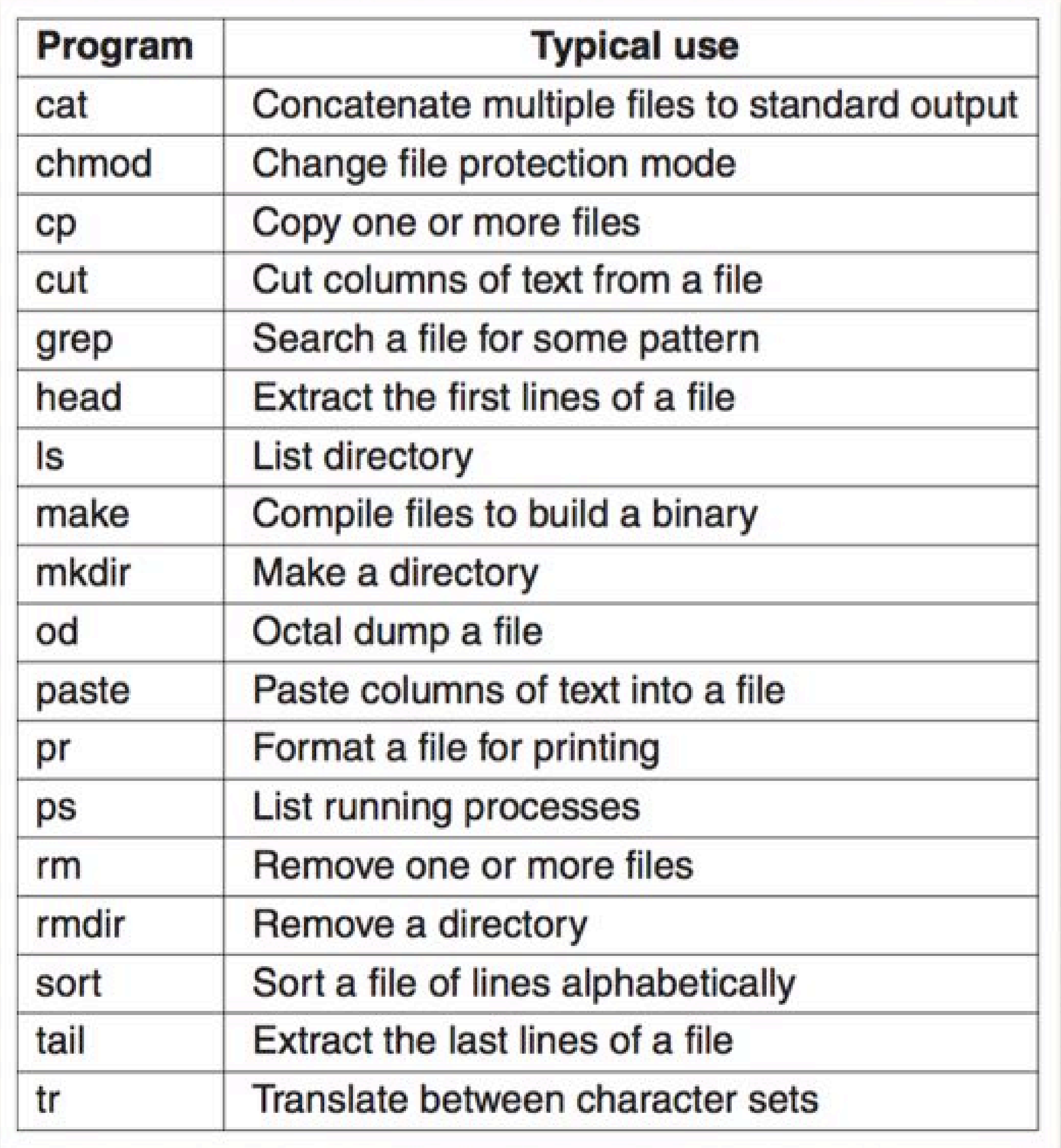

1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2007
2008
2009
2010
2011
2012 to 2013
Open Source
Mixed/Shared Source
Closed Source
Unnamed PDP-7 operating system
Unix Version 1 to 4
Unix Version 5 to 6
PWB/Unix
BSD 1.0 to 2.0
Unix Version 7
Unix/32V
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
Xenix 3.0
BSD 4.2
Sun OS 1 to 1.1
System V R1 to R2
SCO Xenix
Unix Version 8
SCO Xenix V/286
BSD 4.3
AIX 1.0
Sun OS 1.2 to 3.0
System V R3
HP-UX 1.0 to 1.2
Unix 9 and 10 (last versions from Bell Labs)
SCO Xenix V/386
HP-UX 2.0 to 3.0
BSD 4.3 Tahoe
System V R4
SCO Xenix V/386
BSD 4.3 Reno
Minix 1.x
NEXTSTEP/ OPENSTEP 1.0 to 4.0
BSD NET/2
Linux 0.0.1
Sun OS 4
386BSD
HP-UX 6 to 11
NetBSD 0.8 to 1.0
Linux 0.95 to 1.2.x
FreeBSD 1.0 to 2.2.x
BSD 4.4 to 4.4 lite2
SCO Unix 3.2.4
UnixWare 1.x to 2.x
Solaris 2.1 to 9
NetBSD 1.1 to 1.2
OpenBSD 1.0 to 2.2
OpenServer 5.0 to 5.04
NetBSD 1.3
Minix 2.x
FreeBSD 3.0 to 3.2
AIX 3.x to 7.1
Mac OS X Server
OpenServer 5.0.5 to 5.0.7
UnixWare 7.x
Linux 2.0 to 2.6.x
Minix 3.x
Mac OS X 10.0 to 10.9.x (Darwin)
FreeBSD 3.3 to 9.x
NetBSD 1.3 to 6.x
OpenBSD 2.3 to 5.x
OpenServer 6.0
Solaris 10
HP-UX 11i to 11i v3
OpenSolaris and derivatives
Linux 3.x
Solaris 11

MacOS Console



```
Last login: Mon Apr  9 22:06:11 on ttys000
Fabio-MacBook:~ fabio$ man ls
Fabio-MacBook:~ fabio$ ls -la
total 28488
drwxr-xr-x@  114 fabio  staff      3876  8 Apr 13:25 .
drwxr-xr-x     6 root   admin       204 21 Dec  2015 ..
-rw-r--r--     1 fabio  staff        16 14 Jun  2016 .7486160831680234
drwxr-xr-x     6 fabio  staff       204 15 Feb  2015 .AllDRMRemoval
-r--------     1 fabio  staff         7  1 Nov 19:19 .CFUserTextEncoding
-rw-r--r--@    1 fabio  staff     65540  9 Apr 22:33 .DS_Store
drwxr-xr-x     3 fabio  staff       102 15 Feb  2015 .Epubor
-rw-r--r--     1 fabio  staff     45993  1 Jan  2016 .Soulseek.1451672273056
-rw-r--r--     1 fabio  staff     45993  2 Jan  2016 .Soulseek.1451675873387
-rw-r--r--     1 fabio  staff     45993  2 Jan  2016 .Soulseek.1451678942645
drwxr-xr-x     3 fabio  staff       102 18 Apr  2017 .SoulseekQt
drwxrwxrwt@    3 fabio  staff       102  8 Apr  2014 .TemporaryItems
drwx------     6 fabio  staff       204  9 Apr 19:33 .Trash
drwxr-xr-x     3 fabio  staff       102 15 Feb  2015 .Ultimate
-rw-------     1 fabio  staff       327 27 Nov 00:45 .Xauthority
drwxr-xr-x     4 fabio  staff       136 14 Nov  2013 .adobe
drwxr-xr-x     3 fabio  staff       102 25 Feb  2017 .anaconda
drwxr-x---     4 fabio  staff       136 12 Jul  2017 .android
drwxr-xr-x     3 fabio  staff       102  5 Mar  2015 .astropy
drwxr-xr-x    15 fabio  staff       510 22 Nov  2016 .atom
```

Other bash commands



Program	Typical use
cat	Concatenate multiple files to standard output
chmod	Change file protection mode
cp	Copy one or more files
cut	Cut columns of text from a file
grep	Search a file for some pattern
head	Extract the first lines of a file
ls	List directory
make	Compile files to build a binary
mkdir	Make a directory
od	Octal dump a file
paste	Paste columns of text into a file
pr	Format a file for printing
ps	List running processes
rm	Remove one or more files
rmdir	Remove a directory
sort	Sort a file of lines alphabetically
tail	Extract the last lines of a file
tr	Translate between character sets

Tanenbaum p730

the pipe line

- It frequently occurs that the first program in a command line produces output that is used as input to the next program. In the above example, we used the file temp to hold this output. However, Linux provides a simpler construction to do the same thing. In

```
sort <in | head –30
```

- the vertical bar, called the pipe symbol, says to take the output from sort and use it as the input to head, eliminating the need for creating, using, and removing the temporary file. A collection of commands connected by pipe symbols, called a pipeline, may contain arbitrarily many commands. A four-component pipeline is shown by the following example:

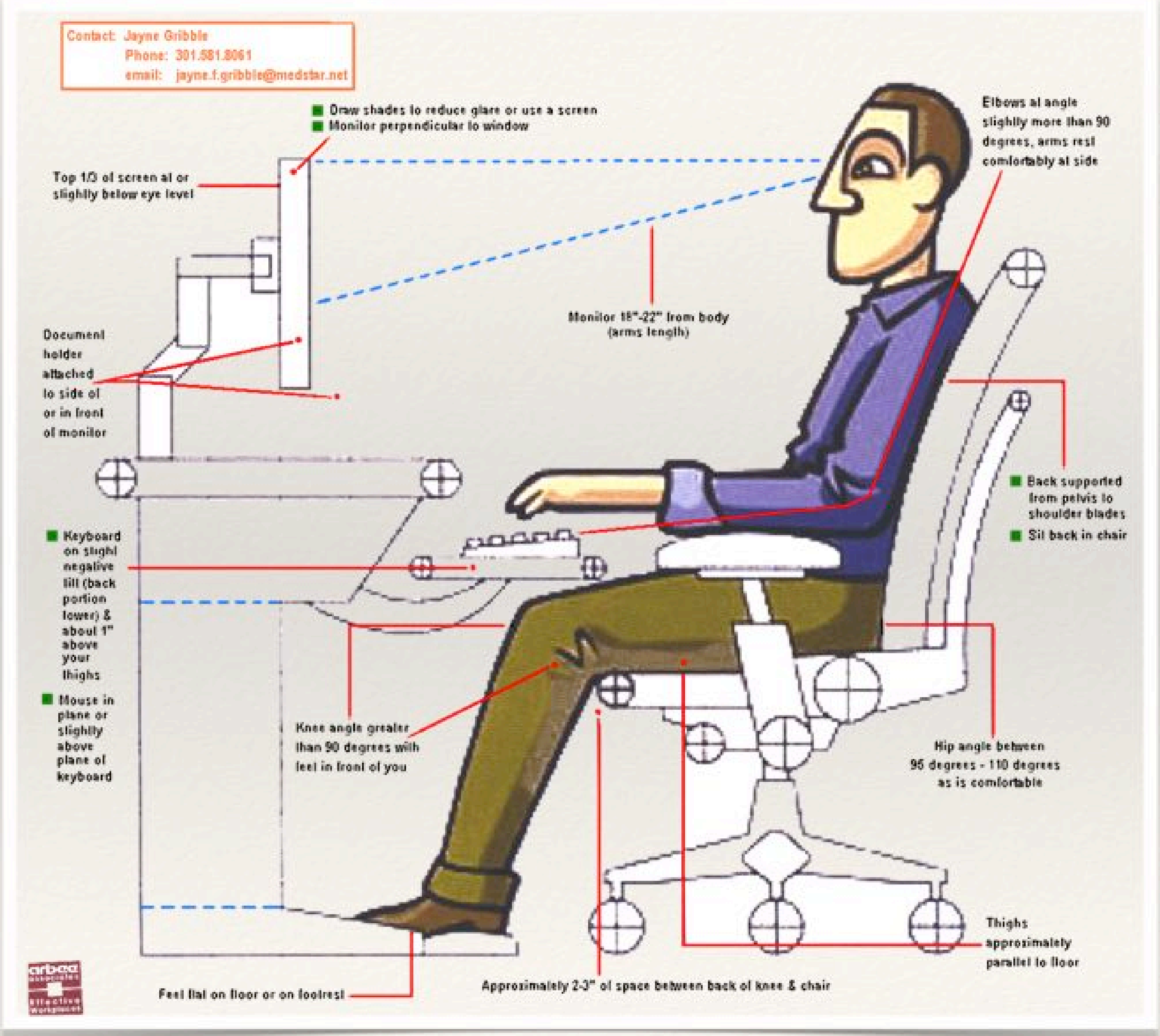
```
grep ter *.t | sort | head –20 | tail –5 >foo
```

- Here all the lines containing the string ‘‘ter’’ in all the files ending in .t are written to standard output, where they are sorted. The first 20 of these are selected out by head, which passes them to tail, which writes the last five (i.e., lines 16 to 20 in the sorted list) to foo. This is an example of how Linux provides basic building blocks (numerous filters), each of which does one job, along with a mechanism for them to be put together in almost limitless ways.

Tanenbaum p728

GUIs

Ergonomics



Contact: Jayne Gribble
Phone: 301.581.8061
email: jayne.f.gribble@medstar.net

■ Draw shades to reduce glare or use a screen
■ Monitor perpendicular to window
Top 1/3 of screen at or slightly below eye level
Monitor 18"-22" from body (arms length)
Elbows at angle slightly more than 90 degrees, arms rest comfortably at side
Document holder attached to side of or in front of monitor
■ Back supported from pelvis to shoulder blades
■ Sit back in chair
■ Keyboard on slight negative tilt (back portion lower) & about 1" above your thighs
■ Mouse in plane or slightly above plane of keyboard
Knee angle greater than 90 degrees with feet in front of you
Hip angle between 95 degrees - 110 degrees as is comfortable
Thighs approximately parallel to floor
Feet flat on floor or on footrest
Approximately 2-3" of space between back of knee & chair

Human-Computer Interaction

- Human-computer interaction (HCI) can be defined in many possible ways.
  - «"Human-computer interaction " is, put simply, the study of people, computer technology and the ways these influence each other. We study HCI to determine how we can make this computer technology more usable by people.»
- The computer is a **tool** that can extend our reach. The design discipline of human-computer interaction systematically applies knowledge about human purposes, human capabilities and limitations, and machine capabilities and limitations in order to enable us to do things that we could not do before. Another goal of HCI, as suggested in the definitions given above, is to **enhance the quality of the interaction between people and computers**. We strive, for example, to make technology **easier for people to learn** and **easier for them to use**.

Baecker p1

Human-Computer Interaction

- Donald A. Norman: **“The Psychology of Everyday Things”** (1988). Examples of bad design, even for objects far simpler than most human-computer interfaces. Norman introduces several concepts that help analyze good and bad design:
  - **Affordances** are the perceived properties of an artifact that determine how it could possibly be used. For example, buttons are for pushing, menus are for choosing.
  - **Constraints** are physical, semantic, cultural, and logical factors that encourage proper actions and prevent erroneous ones.
  - **Conceptual models** are mental models of a system which allow users to understand the system, to predict the effects of their actions, and to interpret the results.
  - **Mappings** describe the relationships between controls and their effects on a system. For example, moving a control to the left should move a corresponding display object left.
  - **Visibility** in the design of a system makes apparent to users the conceptual model of the system and the actions they are allowed to take.
  - **Feedback** from a system provides information about the effects of users' actions.

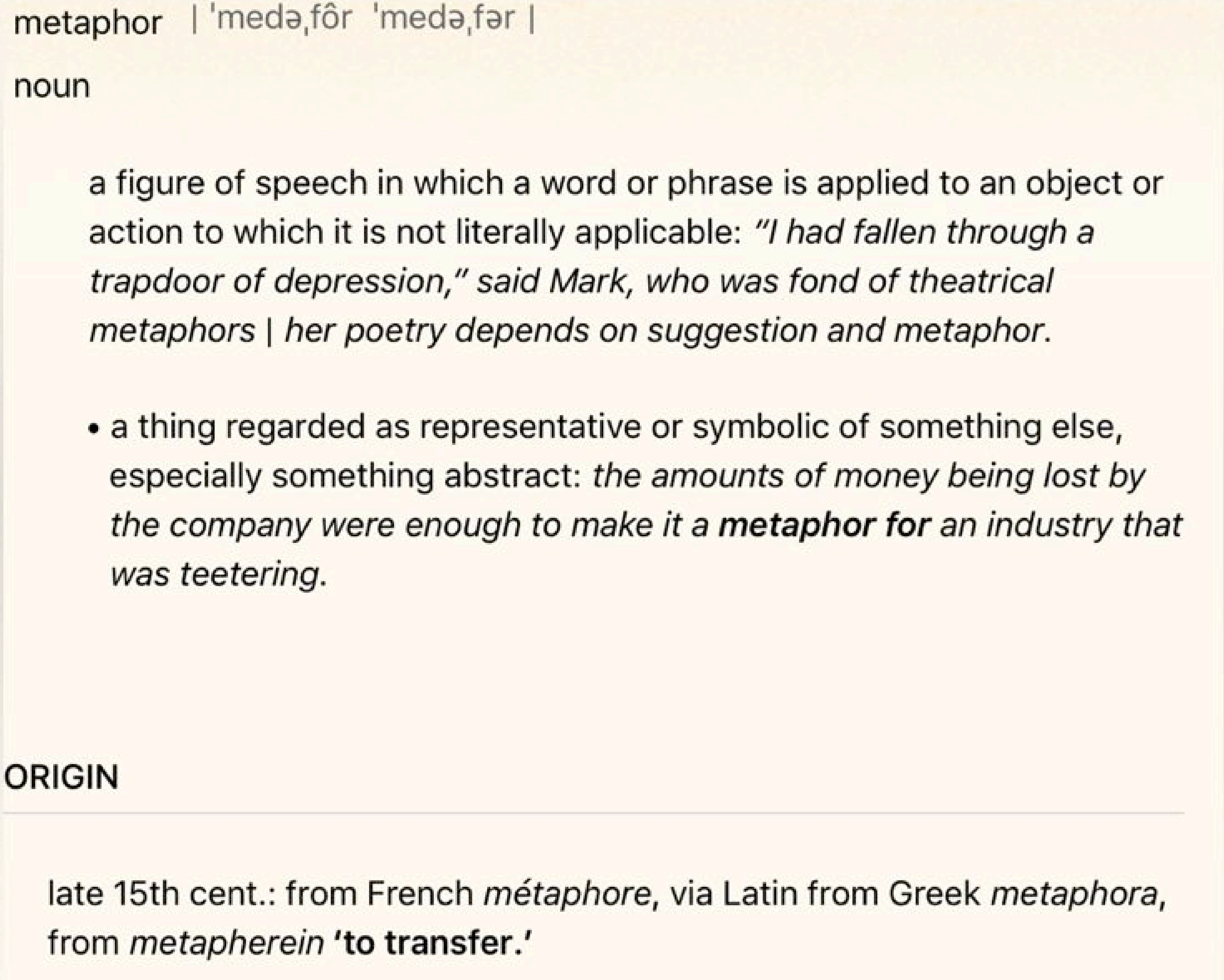
Baecker p1

Human-Computer Interaction

- **Prescriptions for user-centered design**:
  - Make it easy to determine **what actions are possible** at any moment (make use of constraints).
  - Make things **visible**, including the conceptual model of the system, the alternative actions, and the results of actions.
  - Make it easy to evaluate **the current state of the system**.
  - Follow **natural mappings** between **intentions** and e **required actions**; between **actions** and e **resulting effect**; and between the **information** that is visible and the interpretation of **the system state**.
    In other words, make sure that (1) the user can figure out **what to do**, and (2) the user can tell **what is going on**.

Baecker p1

Metaphors



metaphor | ˈmedəˌfôr ˈmedəˌfər |

noun

a figure of speech in which a word or phrase is applied to an object or action to which it is not literally applicable: *"I had fallen through a trapdoor of depression," said Mark, who was fond of theatrical metaphors* | *her poetry depends on suggestion and metaphor.*

- a thing regarded as representative or symbolic of something else, especially something abstract: *the amounts of money being lost by the company were enough to make it a* ***metaphor for*** *an industry that was teetering.*

ORIGIN

late 15th cent.: from French *métaphore*, via Latin from Greek *metaphora*, from *metapherein* **'to transfer.'**

The desktop metaphor

- In computing, the desktop metaphor is an interface metaphor which is a set of unifying concepts used by graphical user interfaces to help users interact more easily with the computer. The desktop metaphor treats the computer monitor as if it is the user's **desktop**, upon which objects such as documents and folders of documents can be placed. A document can be opened into a **window**, which represents a paper copy of the document placed on the desktop. Small applications called desk accessories are also available, such as a desk calculator or notepad, etc.

Wikipedia

Computer Metaphors

- When working on Linux systems through a **graphical interface**, users may use mouse clicks to run applications or open files, drag and drop to copy files from one location to another, and so on. In addition, users may invoke a terminal emulator program, or xterm, which provides them with the basic command-line interface to the operating system.

📖 Tanenbaum p726

Computer Metaphors

- The GUI for Linux is similar to the first GUIs developed for UNIX systems in the 1970s, and popularized by Macintosh and later Windows for PC platforms. The GUI creates a **desktop environment**, a familiar **metaphor** with windows, icons, folders, toolbars, and drag-and-drop capabilities. A full desktop environment contains a window manager, which controls the placement and appearance of windows, as well as various applications, and provides a consistent **graphical interface**. Popular desktop environments for Linux include GNOME (GNU Network Object Model Environment) and KDE (K Desktop Environment).

Tanenbaum p725

Computer Metaphors

DELETE COOKIES?!
gegen-den-strich.com
FERNANDEZ

Open Source Software

Free Software

- The **free software movement** (FSM) or free/open source software movement (FOSSM) or free/libre open source software (FLOSS) is a social movement with the goal of obtaining and guaranteeing certain freedoms for software users, namely the freedom to run the software, to study and change the software, and to redistribute copies with or without changes.
- Regarding the meaning and misunderstandings of the word free, those who work within the free software camp have searched for less ambiguous terms and analogies like "**free beer vs free speech**" in efforts to convey the intended semantics, so that there is no confusion concerning the profitability of free software.
- The two most prominent people associated with the movement, Richard Stallman and Linus Torvalds,

Wikipedia

Richard Stallman

GNU project

Linus Torvalds

Linux project