Computer Systems and Programming

Gang Chen

chengangcs@gmail.com

Outline

- Information in Computer
- Source codes -> Program
- Storage, Operating System and network
- Advanced Topics

Information in Computer

Hello World

Text File

```
#include<stdio>

int main(){
    print("hello world\n");
    return 0;
}
```

Hello World

Text File

```
#include<stdio>

int main(){
    print("hello world\n");
    return 0;
}
```

ASCII Characters

```
> od -t a src/hello.c
```

```
0000000   #   i   n   c   l   u   d   e   <   s   t   d   i   o
0000020   >  nl  nl   i   n   t  sp   m   a   i   n   (   )   {
0000040  sp  sp  sp   p   r   i   n   t   f   (   "   H   e   l
0000060  sp   W   o   r   l   d   \   n   "   )   ;  nl   }  nl
0000076
```

ASCII Characters

```
> od -t a src/hello.c
```

```
0000000   #   i   n   c   l   u   d   e   <   s   t   d   i   o   .   h
0000020   >  nl  nl   i   n   t  sp   m   a   i   n   (   )   {  nl  sp
0000040  sp  sp  sp   p   r   i   n   t   f   (   "   H   e   l   l   o
0000060  sp   W   o   r   l   d   \   n   "   )   ;  nl   }  nl
0000076
```

ASCII Characters

```
> od -t a src/hello.c
```

```
0000000   #   i   n   c   l   u   d   e   <   s   t   d
0000020   >  nl  nl   i   n   t  sp   m   a   i   n   (
0000040  sp  sp  sp   p   r   i   n   t   f   (   "   H
0000060  sp   W   o   r   l   d   \   n   "   )   ;  nl
0000076
```

ASCII Codes

> od -t d1 src/hello.c

```
0000000   35 105 110  99 108 117 100 101
0000020   62  10  10 105 110 116  32 109
0000040   32  32  32 112 114 105 110 116
0000060   32  87 111 114 108 100  92 110
0000076
```

ASCII Codes

```
> od -t d1 src/hello.c
```

```
0000000   35 105 110  99 108 117 100 101
0000020   62  10  10 105 110 116  32 109  9
0000040   32  32  32 112 114 105 110 116 1
0000060   32  87 111 114 108 100  92 110
0000076
```

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Binary

```
> xxd -b src/hello.c
0000000: 00100011 01101001 01101110 01100011 01101100 01110101  #inclu
0000006: 01100100 01100101 00111100 01110011 01110100 01100100  de<std
000000c: 01101001 01101111 00101110 01101000 00111110 00001010  io.h>.
0000012: 00001010 01101001 01101110 01110100 00100000 01101101  .int m
0000018: 01100001 01101001 01101110 00101000 00101001 01111011  ain(){
000001e: 00001010 00100000 00100000 00100000 00100000 01110000  .    p
0000024: 01110010 01101001 01101110 01110100 01100110 00101000  rintf(
000002a: 00100010 01001000 01100101 01101100 01101100 01101111  "Hello
0000030: 00100000 01010111 01101111 01110010 01101100 01100100   World
0000036: 01011100 01101110 00100010 00101001 00111011 00001010  \n");.
000003c: 01111101 00001010
```

Non-ASCII Characters?

Code for Chinese

- GB2312 and GB18030 for Simplified Chinese
- BIG5 for Traditional Chinese
- UTF-8 for everything

GB2312 and GB18030

- GB2312 is the registered internet name for a key official character set of the People's Republic of China, used for simplified Chinese characters. GB abbreviates Guojia Biaozhun (国家标准), which means national standard in Chinese.
- includes 6,763 Chinese characters
- "陈": 1934
- "钢": 2454
- GB18030 is the official character set of the PRC superseding GB2312.
- GBK is a superset of GB2312 developed by Microsoft.

see: GB2312

BIG5

- Big5 is a Chinese character encoding method used in Taiwan, Hong Kong, and Macau for Traditional Chinese characters.
- The original Big5 includes 11151 characters.
- "陳": B3AF
- "鋼": BFFB
- EUC-TW (Extended Unix Code) is another traditional Chinese encoding method, but is seldom used.

One World, One Code

```
My name in simplified Chinese is 陈钢
My name in traditional Chinese is 陳鋼
お名前は何ですか?
당신의 이름은 무엇입니까?
```

What encoding method should be used for this text file?

One World, One Code

```
My name in simplified Chinese is 陈钢
My name in traditional Chinese is 陳鋼
お名前は何ですか?
당신의 이름은 무엇입니까?
```

What encoding method should be used for this text file? Unicode

Unicode

Unicode is a computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems.

UTF-8 is an 8-bit variable-width encoding which maximizes compatibility with ASCII.

UTF-8

- My name in simplified Chinese is 陈钢

  \x4D\x79\x20\x6E\x61\x6D\x65\x20\x69\x6E\x20\x73\x69\x6D\x70\x6C\x69\x66\x69
  \xE9\x99\x88\xE9\x92\xA2

- My name in traditional Chinese is 陳鋼

  \x4D\x79\x20\x6E\x61\x6D\x65\x20\x69\x6E\x20\x74\x72\x61\x64\x69\x74\x69\x6F
  \xE9\x99\xB3\xE9\x8B\xBC

- お名前は何ですか?

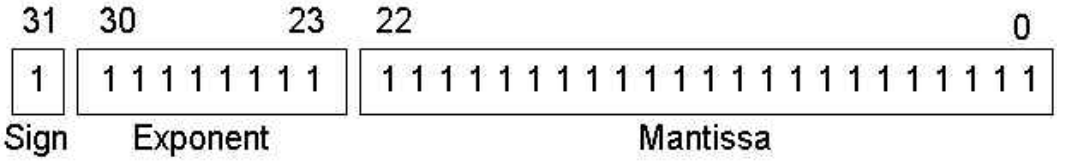
  \xE3\x81\x8A\xE5\x90\x8D\xE5\x89\x8D\xE3\x81\xAF\xE4\xBD\x95\xE3\x81\xA7\xE
  \x3F

- 당신의 이름은 무엇입니까?

  \xEB\x8B\xB9\xEC\x8B\xA0\xEC\x9D\x98\x20
  \xEC\x9D\xB4\xEB\xA6\x84\xEC\x9D\x80\x20
  \xEB\xAC\xB4\xEC\x97\x87\xEC\x9E\x85\xEB\x8B\x88\xEA\xB9\x8C
  \x3F

Number

Floating Point



a = b * 2 ^ c

- Sign is 0 if a is positive, otherwise sign is 1;
- Exponent = 2 ^(n -1) - 1 + c

Floating Point: Example

12.75

1. 12.75 = 1100.11
2. 1100.11 = 1.10011 * 2^3
3.
   - exponent = 2 ^ (8-1) - 1 + 3 = 130 = 10000010
   - sign: 0
   - Mantissa = 10011000000000000000000
4. 12.75 = 0 10000010 10011000000000000000000

   = 414C0000

IEEE 754 Floating Point

- Input: 3.1415926
- Single Precision: 0 10000000 10010010000111111011010, 40490FDA
- Double Precision: 0 10000000000 1001001000011111101101001101000100101101100001001010, 400921FB4D12D84A

IEEE 754 Floating Point

- Input: 3.1415926
- Single Precision: 0 10000000 10010010000111111011010, 40490FDA
- Double Precision: 0 10000000000 1001001000011111101101001101000100101101100001001010, 400921FB4D12D84A
- Single Precision code of 3.1415926 = 3.1415925

How to make the source codes executable?

```
> gcc -o hello hello
> ./hello
hello world
```

Preprocessing

The preprocessor modifies the original C program according to directives that begin with the # character.

```
> gcc -E hello.c > hello.i
```

Compilation

The compiler translates the text file hello.i into the text file hello.s, which contains an assembly-language program.

```
> gcc -S hello.c > hello.s
```

Assembly

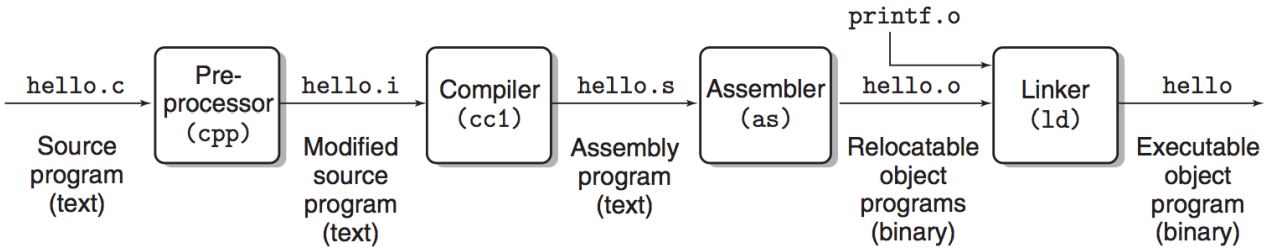
The assembler translates hello.s into machine language instructions.

```
gcc -c hello.c > hello.o
```

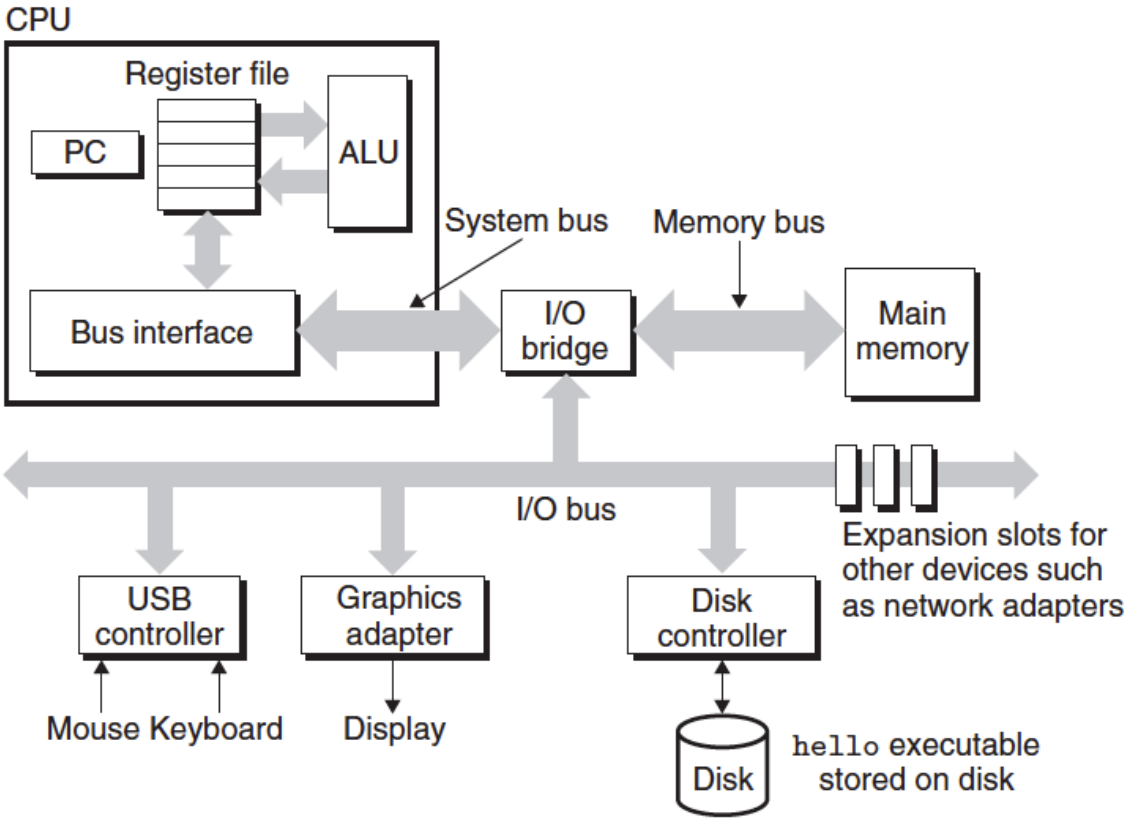
Linking

The linker merge the functions provide by other compiled object files.

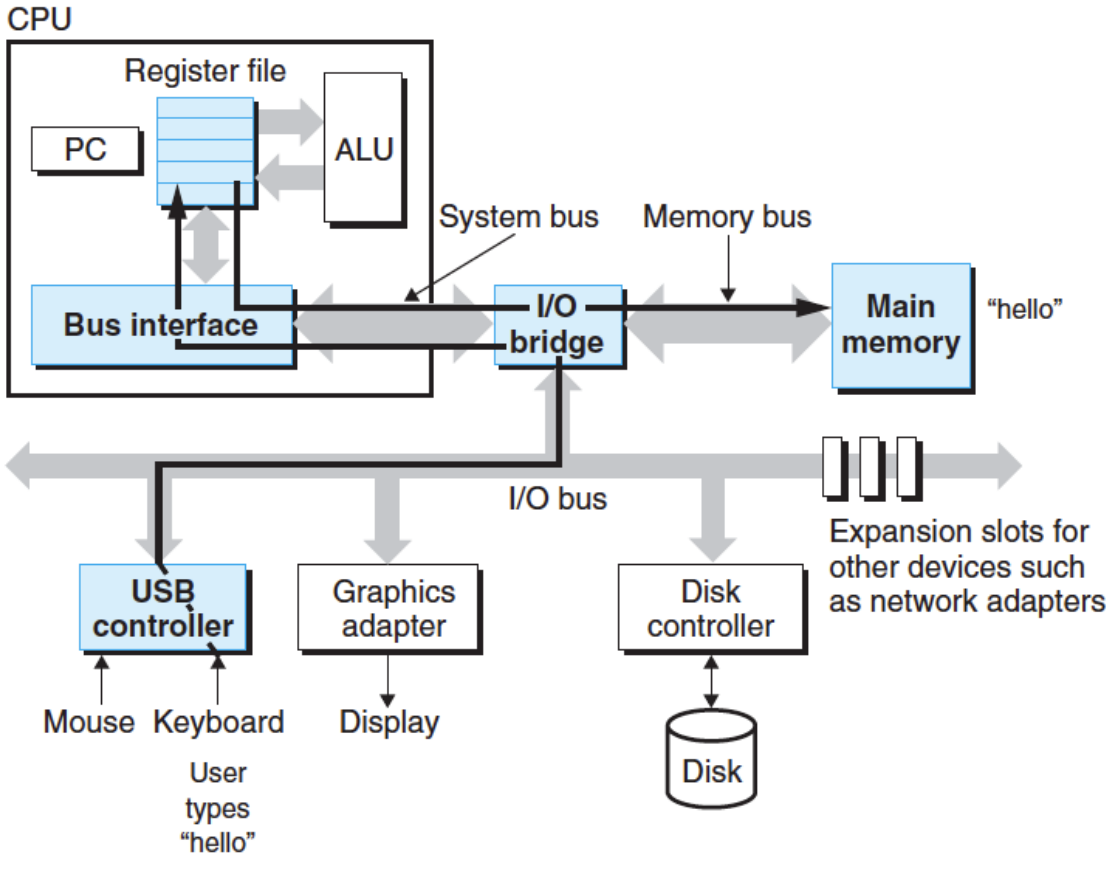
```
> gcc -o hello hello.c
```

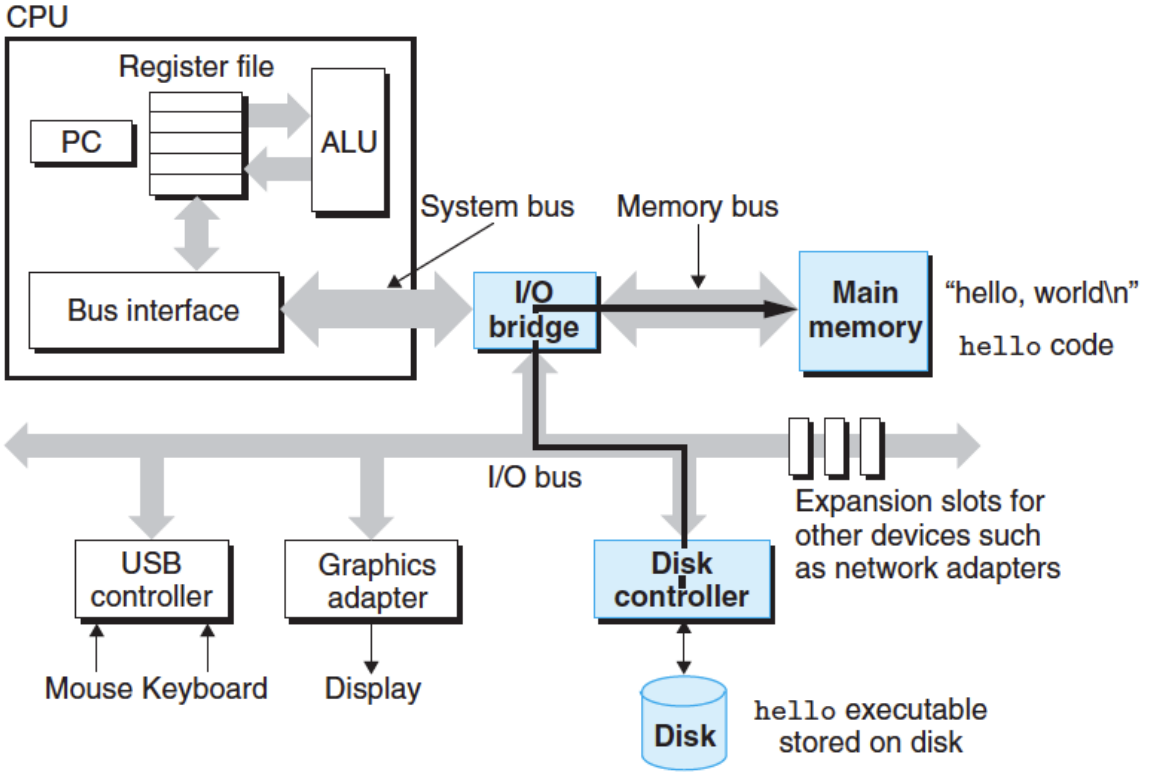

printf.o
hello.c
Pre-
processor
(cpp)
hello.i
Compiler
(cc1)
hello.s
Assembler
(as)
hello.o
Linker
(ld)
hello
Source
program
(text)
Modified
source
program
(text)
Assembly
program
(text)
Relocatable
object
programs
(binary)
Executable
object
program
(binary)

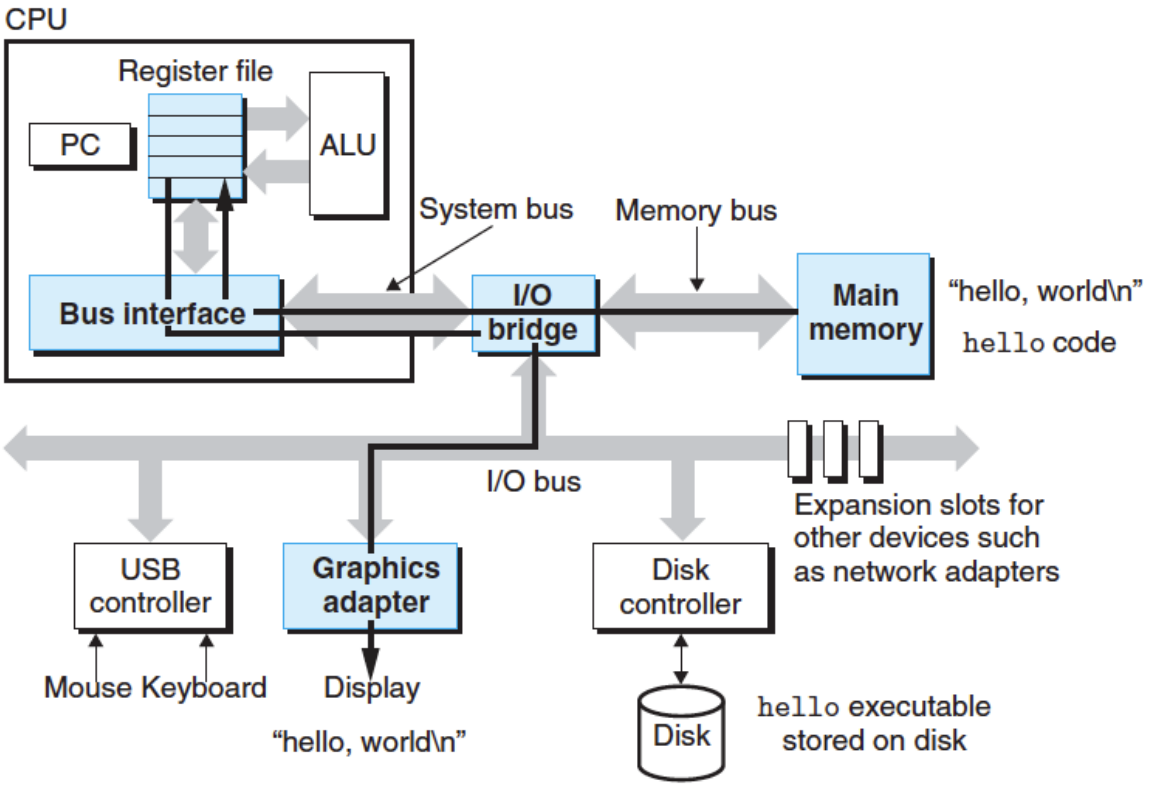
Run the program


CPU
Register file
PC
ALU
System bus
Memory bus
Bus interface
I/O bridge
Main memory
I/O bus
Expansion slots for other devices such as network adapters
USB controller
Graphics adapter
Disk controller
Mouse Keyboard
Display
Disk
hello executable stored on disk

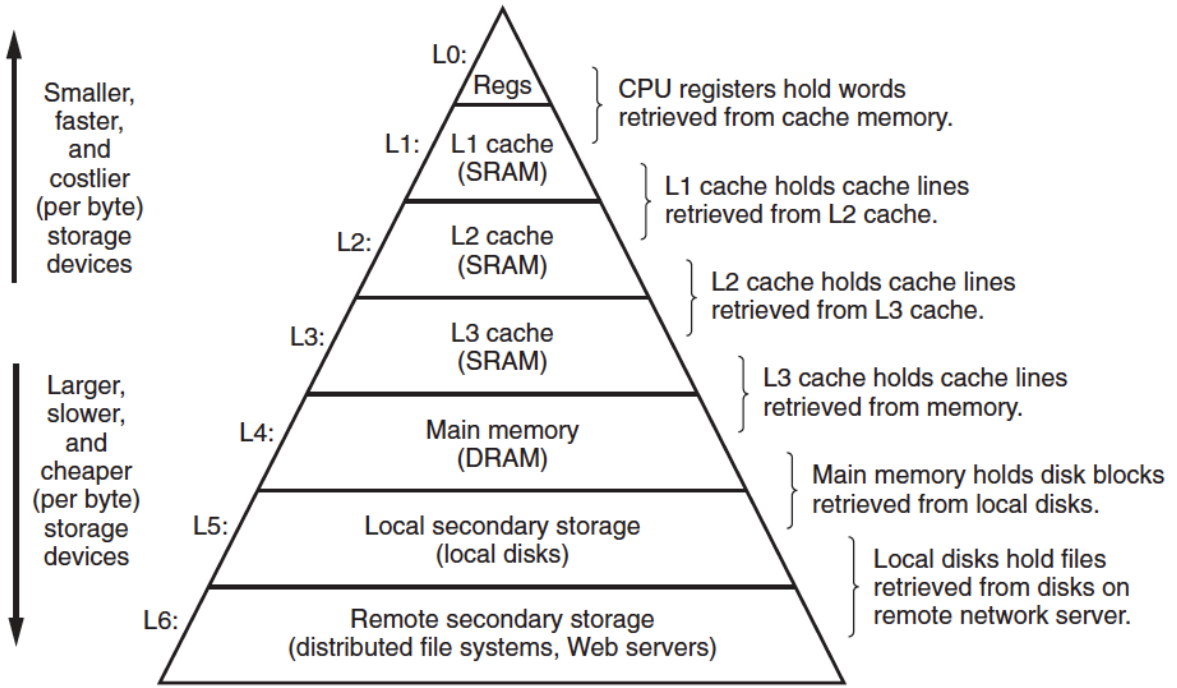
```
> ./hello
hello world
```

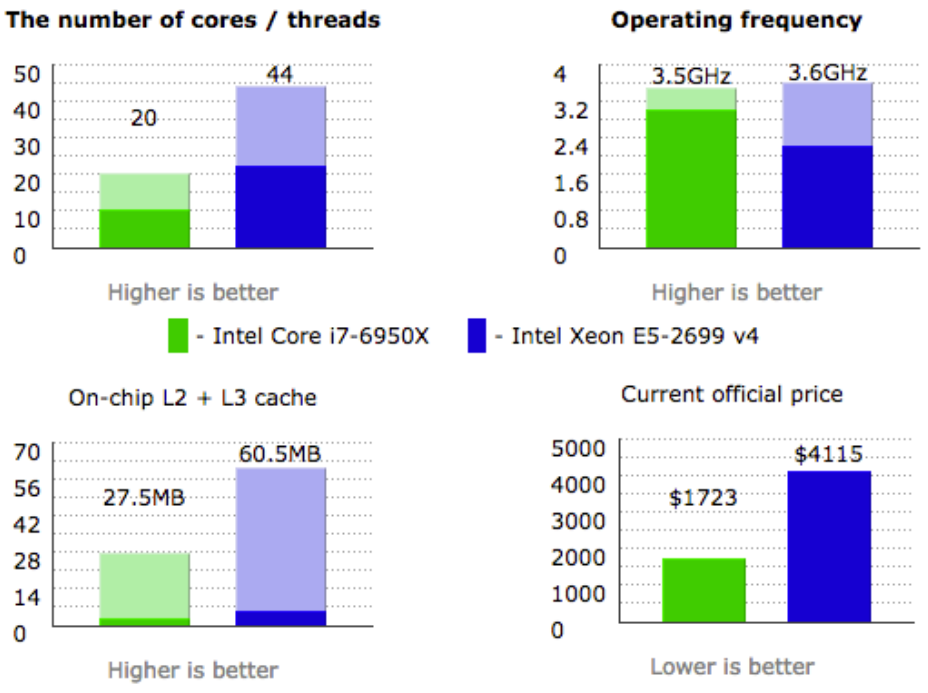

CPU
Register file
PC
ALU
System bus
Memory bus
Bus interface
I/O bridge
Main memory
"hello"
I/O bus
Expansion slots for other devices such as network adapters
USB controller
Graphics adapter
Disk controller
Mouse
Keyboard
Display
Disk
User types "hello"


CPU
Register file
PC
ALU
System bus
Memory bus
Bus interface
I/O bridge
Main memory
"hello, world\n"
hello code
I/O bus
Expansion slots for other devices such as network adapters
USB controller
Graphics adapter
Disk controller
Mouse Keyboard
Display
Disk
hello executable stored on disk

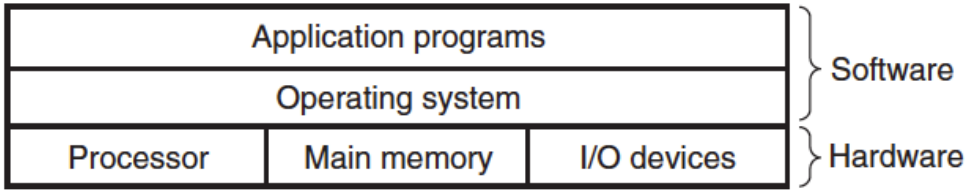

CPU
Register file
PC
ALU
System bus
Memory bus
Bus interface
I/O bridge
Main memory
"hello, world\n"
hello code
I/O bus
Expansion slots for other devices such as network adapters
USB controller
Graphics adapter
Disk controller
Mouse Keyboard
Display
"hello, world\n"
Disk
hello executable stored on disk

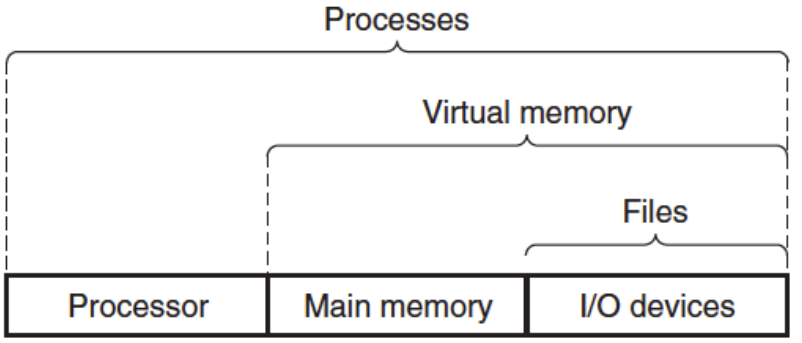
Storage


Smaller, faster, and costlier (per byte) storage devices
Larger, slower, and cheaper (per byte) storage devices
L0:
Regs
CPU registers hold words retrieved from cache memory.
L1:
L1 cache (SRAM)
L1 cache holds cache lines retrieved from L2 cache.
L2:
L2 cache (SRAM)
L2 cache holds cache lines retrieved from L3 cache.
L3:
L3 cache (SRAM)
L3 cache holds cache lines retrieved from memory.
L4:
Main memory (DRAM)
Main memory holds disk blocks retrieved from local disks.
L5:
Local secondary storage (local disks)
Local disks hold files retrieved from disks on remote network server.
L6:
Remote secondary storage (distributed file systems, Web servers)


The number of cores / threads
50
40
30
20
10
0
20
44
Higher is better
Operating frequency
4
3.2
2.4
1.6
0.8
0
3.5GHz
3.6GHz
Higher is better
- Intel Core i7-6950X
- Intel Xeon E5-2699 v4
On-chip L2 + L3 cache
70
56
42
28
14
0
27.5MB
60.5MB
Higher is better
Current official price
5000
4000
3000
2000
1000
0
$1723
$4115
Lower is better

Operating Systems


Application programs
Operating system
Processor
Main memory
I/O devices
Software
Hardware


Processes
Virtual memory
Files
Processor
Main memory
I/O devices

POSIX

The Portable Operating System Interface (POSIX) is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems.

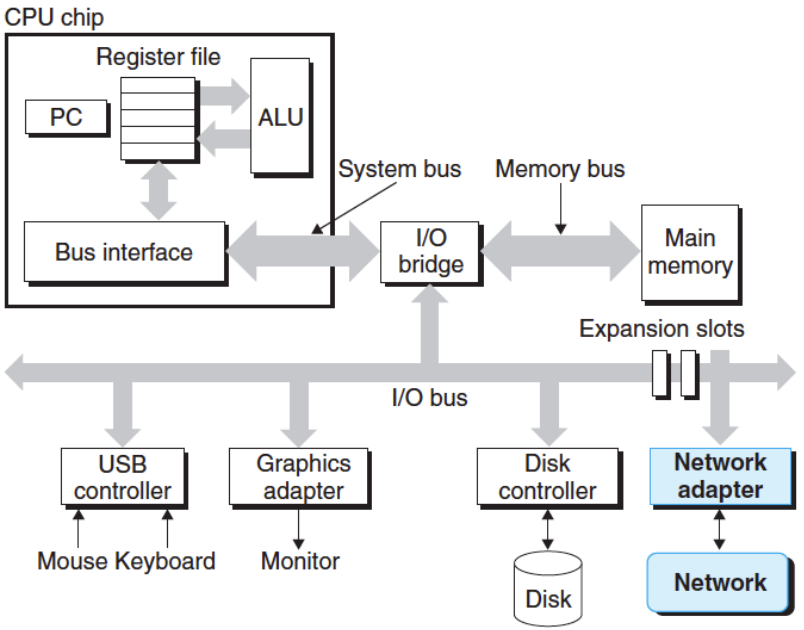
POSIX-certified:

- AIX
- Solaris
- Mac OS X (since 10.5)

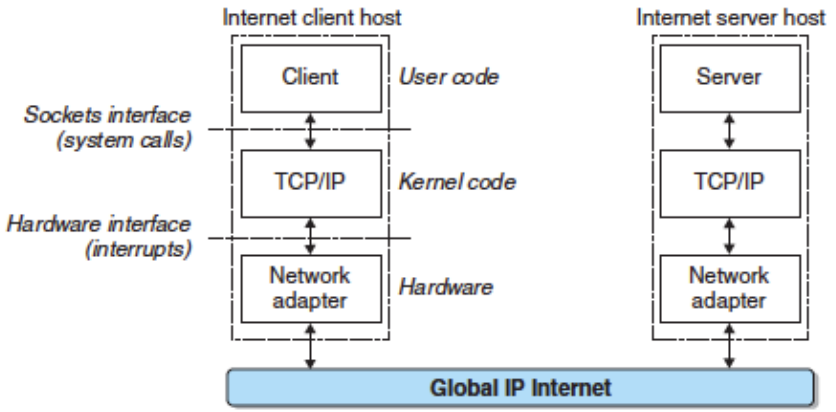
Mostly POSIX-compliant:

- Android
- FreeBSD
- Darwin (core of OS X and iOS)
- Linux
- MINIX
- OpenBSD
- VxWorks

Network


CPU chip
Register file
PC
ALU
System bus
Memory bus
Bus interface
I/O bridge
Main memory
Expansion slots
I/O bus
USB controller
Graphics adapter
Disk controller
Network adapter
Mouse Keyboard
Monitor
Disk
Network

Internet


Internet client host
Internet server host
Client
User code
Server
Sockets interface
(system calls)
TCP/IP
Kernel code
TCP/IP
Hardware interface
(interrupts)
Network
adapter
Hardware
Network
adapter
Global IP Internet

HTTP

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP functions as a request–response protocol in the client–server computing model.

Advanced Topics

- Concurrency
- Parallelism
- Distributed
- Quantum computing