Introduction

010.133
Digital Computer Concept and Practice
Spring 2013

Lecture 01

Instructor

- Prof. Jaejin Lee
  - Email: jlee@cse.snu.ac.kr
  - URL: http://aces.snu.ac.kr/~jlee
  - Office: Room 505, Building 301
  - Office hours: Mon. and Wed. 10:00AM – 11:00AM
  - Phone: 880-1863

TAs

- Jeongho Nah
  - Email: jeongho@aces.snu.ac.kr
  - Office: Room 312, Building 138
  - Office hours: TBD
- Jinyoung Joo
  - Email: jinyoung@aces.snu.ac.kr
  - Office: Room 520, Building 301
  - Office hours: TBD

Textbook

- Lecture notes and slides
- Handouts

Attendance and Course URL

- Attendance
  - Students are required to attend class regularly
  - Attendance will be recorded and will affect your grades directly or indirectly
  - The decision is up to the instructor
- Course URL
  - http://etl.snu.ac.kr

Assignments (tentative)

- There will be approximately 10 homework and programming assignments
- At the beginning of the semester, each student has a total of 3 grace days that can be used as extension days for any assignments ***other than the last assignment***
- You can use all 3 days on one assignment or split them up across two or three assignments
- After you use all your 3 grace days, the late submission will not be accepted, and you will get a 0 on the assignment

Exams and Academic Integrity

- There will be an in-class midterm exam and a final exam
  - The exam time and locations will be announced later
- All assignments must be done from scratch
- The solutions of the problems must be your own work
- Any sort of cheating is not allowed
- We expect all students to adhere to Seoul National University's school regulations on integrity of scholarship and grades

Grading (tentative)

- Final grades will be based on the following:
  - 10% Class attendance and participation
  - 30% Homework and programming assignments
  - 30% Midterm Exam
  - 30% Final Exam

Other Policies

- The course website will reflect all modifications
  - The instructor and TAs will use the website to notify you of important changes
  - You are responsible for checking your email and the website regularly
- Failure to take an examination at the scheduled time will result in a 0 for the examination except in the cases of documented emergency
  - You should discuss with the instructor any extenuating circumstances that impact on your participation in the course as soon as those circumstances are known

Introduction

What is a Computer?

- An electronic device that accepts input, stores data, processes data according to a set of instructions (called program), and produces output in desired form
  - Abstracted as a black box that accepts input and produces output
    - The output depends on the current program in the block box

Input, Output, and Data

- Input
  - The information that is submitted to a computer by a human, by another computer, or by its environment
- Output
  - The result produced by the computer
  - Texts, audio, graphs, pictures, etc.
- Data
  - Factual information in a form suitable for use with a computer
  - Stored in the computer in the form of electrical signals
    - Have two states represented by 0 or 1

Bits, Bit Patterns, and Bytes

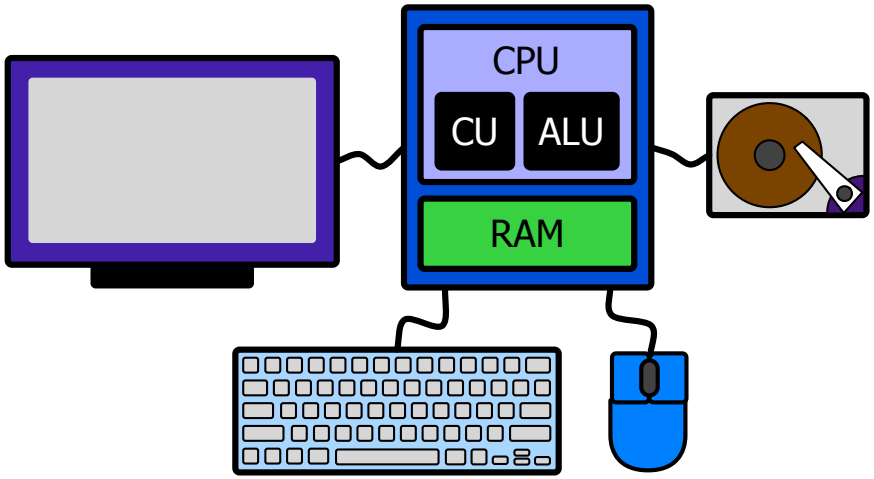
- A bit (a contraction of binary digit) is the fundamental unit of information in computer science
  - Two possible distinct values, 0 or 1
  - Inside a computer, data is encoded as patterns of bits
    - 0s and 1s
- A bit pattern is a sequence (or string) of bits
  - The meaning of a bit pattern depends on the interpretation
    - Numeric values
    - Symbols such as characters in an alphabet
    - An image
    - Audio
    - …
- A byte is another unit of information
  - consists of 8 bits
  - Historically, a byte was used to encode a single character of text

Hardware and Software

- Hardware is the physical components of the computer system
- Software is a set of programs that instructs the hardware to obtain the output
  - A computer program is a set of instructions telling the computer what to do with the input in order to produce the output

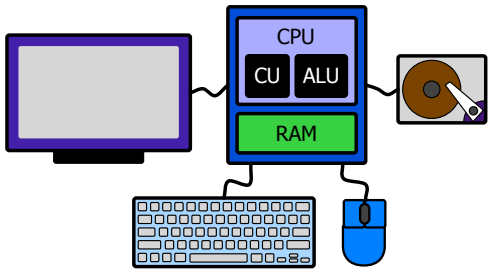
Von Neumann Architecture

- Four basic hardware components
  - Input devices
  - Output devices
  - Main memory
  - Central processing unit (CPU)



Von Neumann Architecture (contd.)

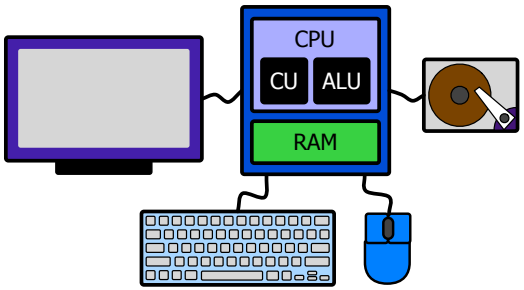
- Input devices
  - Keyboards, mice, hard disk drives, bar code readers, etc.
  - Transmit information from the outside world into main memory
- Output devices
  - Screens, printers, hard disk drives, etc.
  - Transmit information from main memory to the outside world



- Main memory (primary storage)
  - Stores both the program and the data being processed
  - RAM (random-access memory)
    - The location of data does not affect the access speed
  - Typically volatile
    - When the power is turned off, the information stored in main memory is lost

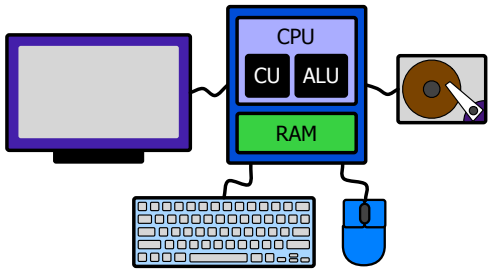
CPU

- Machine code
  - The representation of a program that is actually read, interpreted, and executed by the computer
  - A program in machine code consists of a sequence of machine instructions
    - A machine instruction is represented as a finite bit string
- CPU carries out the instructions of a computer program
  - Arithmetic logic unit (ALU) performs arithmetic and logical operations
  - Control unit (CU) fetches instructions from main memory, decodes them, and executes them
    - Uses the ALU to execute the instructions when necessary



Secondary Storage

- Alternatively referred to as auxiliary storage or external memory
- Hard disk drives, solid-state disk drives, flash memory (e.g., USB sticks), floppy disks, magnetic tape, punched cards, and paper tape
- Non-volatile
  - It does not lose data stored when the power is down
- Used as both input and output devices
- For storing programs and data
- The computer usually accesses it through an intermediate space in main memory

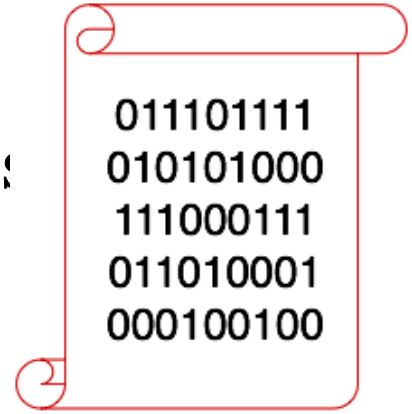


Stored Program Concept

- Conceptually, programs and data are very different
- But, we treat programs as data
  - Both can be stored in main memory
- The program is easily replaced by another program for a different task
  - The program for a specific task is loaded into main memory (e.g., from secondary storage), and instructions in the program are executed one after another without any human intervention
  - The hours of tedious labor required to reprogram computers can be eliminated
- A modern computer can solve almost an infinite variety of problems
  - By just switching between different programs

Files

- A file is a complete collection of data treated by a computer as a unit especially for purposes of input and output (Merriam-Webster Dictionary)
- The contents of a file are encoded in a sequence of bits
  - The meaning of the bits totally depends on the interpretation by the program that accesses the file
- Text files
  - Contents are interpreted as character symbols
- Binary files
  - Other files than text files
  - A binary file contains any type of data encoded in bits



ASCII

- A character is encoded as a bit string in a text file
- ASCII (American Standard Code for Information Interchange)
- Developed by ANSI (American National Standards Institute)
- The most common character coding scheme for English-language text files
- Uses 7 bits for each character symbol
  - 128 (27) different character symbols
  - Each byte in an ASCII text file contains a single character

Dec	Bin	Char	Dec	Bin	Char	Dec	Bin	Char	Dec	Bin	Char
0	0000000	NUL	32	0100000	SPC	64	1000000	@	96	1100000	`
1	0000001	SOH	33	0100001	!	65	1000001	A	97	1100001	a
2	0000010	STX	34	0100010	"	66	1000010	B	98	1100010	b
3	0000011	ETX	35	0100011	#	67	1000011	C	99	1100011	c
4	0000100	EOT	36	0100100	$	68	1000100	D	100	1100100	d
5	0000101	ENQ	37	0100101	%	69	1000101	E	101	1100101	e
6	0000110	ACK	38	0100110	&	70	1000110	F	102	1100110	f
7	0000111	BEL	39	0100111	'	71	1000111	G	103	1100111	g
8	0001000	BS	40	0101000	(	72	1001000	H	104	1101000	h
9	0001001	TAB	41	0101001	)	73	1001001	I	105	1101001	i
10	0001010	LF	42	0101010	*	74	1001010	J	106	1101010	j
11	0001011	VT	43	0101011	+	75	1001011	K	107	1101011	k
12	0001100	FF	44	0101100	,	76	1001100	L	108	1101100	l
13	0001101	CR	45	0101101	-	77	1001101	M	109	1101101	m
14	0001110	SO	46	0101110	.	78	1001110	N	110	1101110	n
15	0001111	SI	47	0101111	/	79	1001111	O	111	1101111	o
16	0010000	DLE	48	0110000	0	80	1010000	P	112	1110000	p
17	0010001	DC1	49	0110001	1	81	1010001	Q	113	1110001	q
18	0010010	DC2	50	0110010	2	82	1010010	R	114	1110010	r
19	0010011	DC3	51	0110011	3	83	1010011	S	115	1110011	s
20	0010100	DC4	52	0110100	4	84	1010100	T	116	1110100	t
21	0010101	NAK	53	0110101	5	85	1010101	U	117	1110101	u
22	0010110	SYN	54	0110110	6	86	1010110	V	118	1110110	v
23	0010111	ETB	55	0110111	7	87	1010111	W	119	1110111	w
24	0011000	CAN	56	0111000	8	88	1011000	X	120	1111000	x
25	0011001	EM	57	0111001	9	89	1011001	Y	121	1111001	y
26	0011010	SUB	58	0111010	:	90	1011010	Z	122	1111010	z
27	0011011	ESC	59	0111011	;	91	1011011	[	123	1111011	{
28	0011100	FS	60	0111100	<	92	1011100	\	124	1111100	|
29	0011101	GS	61	0111101	=	93	1011101	]	125	1111101	}
30	0011110	RS	62	0111110	>	94	1011110	^	126	1111110	~
31	0011111	US	63	0111111	?	95	1011111	_	127	1111111	DEL

Unicode

- Developed by the Unicode Consortium
- A character coding scheme to assign a unique value to every character symbol used in every language in the world
- Texts from multiple languages to appear in a single text file
- Unicode uses 16 bits (2 bytes) for each character symbol
  - 65,536 (2^{16}) different symbols
  - Different sections of Unicode are allocated to character symbols from different languages

Application Software

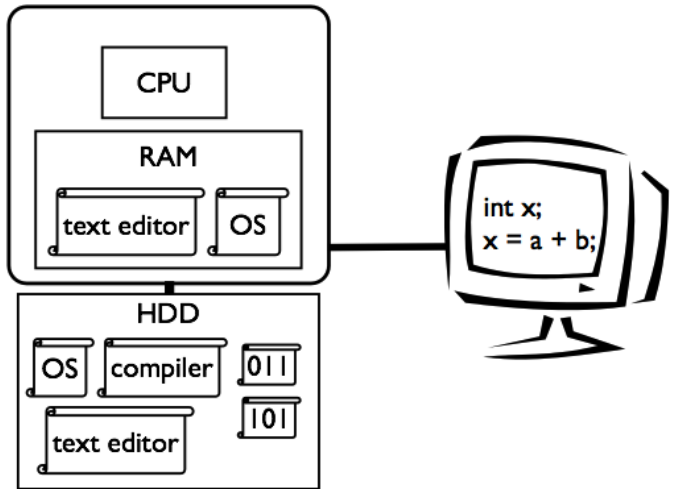
- Also known as an application
- A set of programs that helps the user to carry out a specific task
  - E.g., a spreadsheet

System Software

- A set of programs designed to operate the computer hardware and to provide a platform for running applications
- Operating system
  - Also known as an OS
  - System software that controls the operation of a computer and directs the processing of programs
  - Manages computer hardware resources
  - Provides common services for application software
  - Typically, cannot run an application without an OS
  - Microsoft Windows, Mac OS, Linux, Unix, etc.

Utility Software

- Also referred to as utility, tool, and service program
- System software designed to help the user manage and tune the computer hardware and software
- Focuses on how the computer hardware and software operates
- Virus scanners, data compression utilities, disk partition utilities, archive utilities, system monitors, text editors, assemblers, etc.
- Text editors
  - A utility program for creating and modifying text files
  - GNU emacs, UNIX vi, Microsoft word, etc.



Programming Languages

- A programming language is a formal language in which computer programs are written
- The level of abstraction from the details of the underlying computer in a high-level programming language is higher than a low-level programming language
- High-level programming languages
  - Java, C, C++, FORTRAN, Scheme, ML, etc.
  - More close to natural languages
  - More understandable than a low-level language
  - Makes the process of developing programs simpler and easier

Programming Languages (contd.)

- Low-level programming languages
  - Assembly language
  - Represents machine instructions symbolically
    - There exists an assembly instruction that corresponds to a machine instruction, but not vice versa
  - A program called an assembler is used to translate assembly language instructions into the target computer's machine code instructions
- The definition of a particular programming language consists of both syntax (how the various symbols of the language are combined) and semantics (the meaning of the language constructs)
  - The syntax and semantics of a programming language are typically defined in its specification

C Language

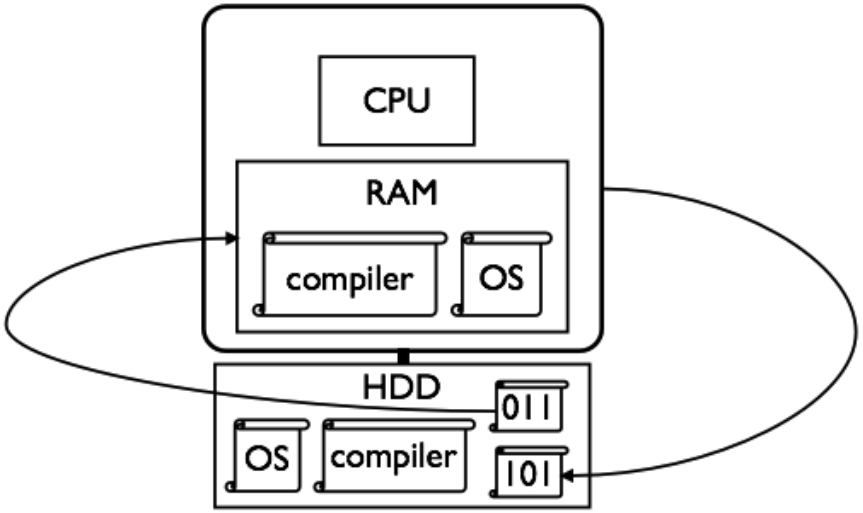
- Developed from 1969 to 1973 by Dennis Ritchie of Bell Laboratories
- Designed for a practical purpose to implement the UNIX operating system
- One of the most widely used programming languages in these days
- ANSI C99 is an internationally recognized C language specification
  - Almost all C compilers follow this standard

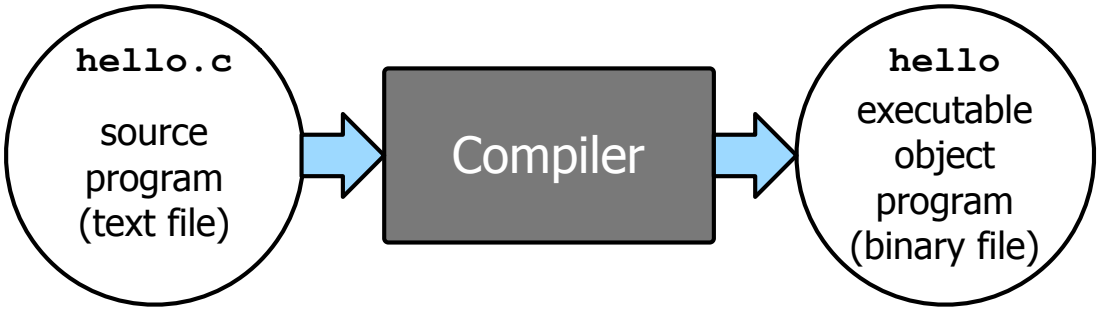
Compilers and Compilation Process

- A compiler is a program that automatically translates another program from some programming language to machine code

```
#include <stdio.h>

int main()
{
    printf("hello, world\n");
}
```





Abstractions in Computer Science

- Computer science is fundamentally a science of abstraction
- Abstraction is the process of considering the external properties of an object independently of its internal details
  - Interested in ”what the object does” without any interest in ”how the object does it”
- To solve a complex problem,
  - Devise an understandable model for it through abstraction
  - Then, explore appropriate methods to solve it using the model