L1

1. assembly language

Instruction set

Micro architecture

Digital logic

2. CPU=center processing unit

ALU=Arithmetic logical unit

3. input, output, cpu, memory

4. mainframe

Supercomputer

Workstation

Pc

5. network

Web

Cloud

6. what is going on inside the computer

Program

7. program written on old hardware can run on new hardwares

8. VHDL= very high speed integrated circuit hardware definition language

9. non-professional easy to understand

Designer easy to design

Hardware engineer easy to maintain

10. the capacity will be double in 24 months

11. operating system

12. ease of programming

Protection for system and user

Fairness and efficiency of system

13. windows, icon, menu, pointer

14. computation

15. data

16. interaction

L2

1. input data -> process ->output data

2. input-process-output

3. program/software

4. hardware, software and data that being manipulated

5. cpu, mouse, keyboard, hard disk, screen

6. cpu

7. main memory/ system memory

8. hard disk/ CD/ DVD

9. keyboard/ scanner/ mouse

10. screen/ speaker/ printer

11. a collection of program

12. von Neumann model is the process part of input-process-output model

13. true

14. false

15. more convenient for programming

User friendly

16. data manipulation

Input and output

hardware and program control

Machine control

17. cpu can only understand binary/machine codes

18. input device -> processor/memory ->output device

19. data and program

20. where: cpu and main memory

cause: cpu fetch speed is faster than main memory

21. the cpu know where to fetch data and program (instructions)

22. Harvard architecture separate data from programs

23. increase the data transfer rates

24. it needs two separated reservoirs for data and program

25. Harvard architecture

L3

1. Java C C++ Python
2. True
3. Compiling interpreting assembling
4. Compiler: HLL ->machine codes

Assembler: machine codes ->binary codes

Interpreter: HLL ->machine codes (on-the-fly)

1. Go back to the source program and debug
2. Join together all the binary codes/ report errors
3. False
4. True
5. A program is separated into different module
6. True
7. intermediate code
8. binary machine code
9. when frequently change the program/ when program run fast
10. it decoded an instruction at one time like machine circle
11. source-level subroutine, macro libraries

pre-translate and re-locate binary libraries

dynamic libraries and linking

1. who own/maintain it
2. true
3. have a private copy of subroutines, wasting memory and swapping time in a multitasking system
4. true

L4

1. a bit is the most basic unit of information
2. true
3. two states: on/off technology
4. true (except 0 and 1)
5. 16
6. 8
7. Shorthand of the binary code/ a hexadecimal number represents 4 binary number
8. 7

L5

1. C
2. B
3. A
4. Printing/ control character
5. IEEE 754 standard
6. The compiler reserve the right amount of space to hold the variable/ decoding scheme

L6

1. Windows interface (WIMP)
2. Resource improperly managed and multiple process and deadlock/ protect the hardware from manual improper operation
3. Programs run simultaneously if they are not too resource consuming
4. Memory management/ security/ allocate the time of cpu processing for each program
5. A system which support multiple users run program at same time
6. Via networking API
7. Phone, fax, computer

L7

1. General purpose machine: can do everything but in low efficiency

Special purpose machine: can do and only can do one thing in high efficiency

1. 6\*(6-1)/2\*(8+2)=150
2. IP(instruction pointer) and IR(instruction register)
3. 1024\*768
4. Math and graphics
5. Data, address and control
6. The fetching speed does not match between cpu and main memory

A bus one item of data each time

1. ALU (arithmetic logic unit)

Control unit

1. Fetch -> decode -> execute -> store the result
2. Parallel port/USB port
3. C
4. C
5. B
6. IP,IR,MAR,MBR

L8

1. False
2. Binary machine codes
3. Memory address
4. True
5. True
6. Put the address of ebx in ebx
7. Jump to the L2 if the value is not zero/ when L2 is missing
8. Illegal, because one cannot move value from memory to memory

L9

1. Zero/sign/overflow flag
2. True
3. No register
4. Stack
5. True
6. False
7. False

L10

1. False 1-15
2. Operation/ the location of operands and results/ the data type
3. Distinct bit fields
4. C
5. True
6. False

L11

1. False
2. False
3. True
4. %s
5. Add esp,12;
6. False

L12

1. True
2. True
3. If not equal loop label
4. If eax is not equal to ebx, then loop label

L13

1. True
2. True
3. As many as you want
4. As many as you want
5. ‘ret’ instruction
6. The CALL instruction will record the current value of EIP

Places the required subroutine into EIP

L14

1. When you only need value
2. When you need the address of parameter
3. Because nested calls
4. Parameters of subroutine

Return address

EBP

Local variables

L15

1. True
2. True
3. Stack modified

Eax original content will be replaced

1. True

L16

1. Cons: less economical/ difficult to calculate

Pros: easy to translate to character

1. Yes
2. Yes
3. Cons: at the point view of humans, difficult to understand

Pros: easy to complement based on 2-state technology

L17

1. False
2. The sign of output is different
3. Yes
4. True
5. 16
6. 64
7. False

L18

1. Sign:1

Exponent:8

Mantissa:23

1. No need
2. When exponent=128
3. 127
4. 10232

L19

1. C
2. B
3. A

L20

1. A
2. A
3. Cache, ram, disk
4. True

L21

1. C
2. B
3. B
4. Abd