

Computer Architecture Assignment : 1

Q.1)

What is Toolchain? Elaborate working of every components of Toolchain i.e. Editor, pre-processor, compiler, assembler, linker, loader.



Toolchain is a process that tells us how the program is get executed and run on the system in a stepwise manner as name suggests toolchain have some tools that are connected to each other and dependent to each other do to their work. It's a chain like structure. Toolchain describes the whole working of program from starting to end.

i) Editor :-

- Editor is a tool in which we write our code edit it and execute it
- Editor provide us environment to write code, edit code & run the code.

In windows Operating editors are :-
Notepad, Visual Studio code, Dev++ ,
Notepad ++, TurboC++, pycharm

In linux o.s. editors are :-

g k write
g edit

In mac o.s. editors are:

Pager, Sublime, X-code, Text edit

2) Preprocessor :-

When we execute the code it goes to the Preprocessor firstly. Preprocessor takes demo.c file as input and gives us demo.i as output.

Where, i = intermediate code.

It converts the code into intermediate code.

Preprocessor have 4 types of work.

- 1) Header file substitution
- 2) Macro expansion
- 3) conditional compilation
- 4) line control

Preprocessor expand the program because it added header file declarations in to it.

3) Compiler :-

Compiler is a tool that takes demo.i file as input from preprocessor and gives demo.asm file as output

Where, asm = assembly code.

- When program is compiled on windows o.s then it gives demo.asm file as output

- when program is in linux os it gives demo.s file as output
- assembly code is a processor dependent it's not run in another processor
- In assembly code there are some mnemonics like PUSH, ADD, RETN, MOV

4) Assembler :-

Assembler takes a demo.asm file as input and gives demo.obj file as output.

↳ Object

It takes demo.asm file from compiler

- Assembler converts the code in binary form like obj

5) Linker :-

Linker takes demo.obj file as input from Assembler and gives demo.exe file as output.

where, exe = executable file

- The main work of linker is it add the ~~demo~~.obj file into demo.obj other and gives demo.exe file as output
- other.obj file adds the header file definition into the program that are needed to run the program

Q) Loader :-

Loader takes the demo.exe file from system harddisk and bring it to the RAM

for ~~ex~~ running the program it necessary to bring it on the RAM loader bring it to the RAM for run

Q) What are the different tasks of operating system?



Operating system do not work by itself it do their work by the following

- 1) process manager
- 2) file manager
- 3) memory manager
- 4) CPU Scheduler
- 5) Hardware Abstractor

1) process manager :-

All process related to work handing by the process manager

2) file manager :-

- All files related to work handing by the file manager

3) memory manager :-

The memory related to work handing by the memory manager

4)

CPU Scheduler :-

All CPU related works are handle by the CPU scheduler

5)

Hardware Abstractor :-

all hardware related work handle by the hardware abstractor

Operating system have 2 types of executable file.

- 1) self executable - executed by self
- 2) dependent executable - depend on other to execute

Q.3) What primary Header contains? Explain in details.

The Hello.exe file divided into 4 parts PTH, Text, data, symbol table.

PH = Primary Header

PH stores the four things

- i.e.
- 1) Magic number
 - 2) address of entry point function
 - 3) type of executable
 - 4) time stamp

1) Magic number :-

- every operating system have it's magic number to identifying it

- PH stores the magic number.

- operating system magic number should be same to the PH magic number to execute the program

- 2) Address of entry point function :-
- In PH in second position it stores the entry point function.
- It tells us from which line execute the code.

- 3) Type of executable :-
- PH third part stores the type of executable.
- there are 2 types of executable file
 i) self executable
 ii) dependent executable

- .exe and .cPA are self executable file for windows.
- For linux .o file is self executable
- For mac .dmg file is self executable

- 4) Time stamp :-
- It tells us the time & date of execution.

Q4] What is your

Q4] Explain in details text, data, symbol table and stack section

1) text :-

It stores the binary code that is in obj form. Stores binary instructions

2) Data :-

Data section store the memory for the global variable only.

It has two parts

- 1) BSS
- 2) Non BSS.

1) BSS :-

- It stands for Block Starting with Symbol
- It stores the memory for non-initialize global variable.

2) Non BSS

- It stands for Block starting with value.
- It stores the memory for ~~non~~ initialized global variable.

3) Symbol Table :-

which contain information about variables & symbols

- symbol table is a metadata that means data about data.
- symbol table have 4.6 entries

4) Stack section :-

- stores the memory for local variables when function call the function frame memory comes in stack

Q. 5) Explain the working of ECU, Ig, RCU?



ECU :-

Execution Control Unit

It has 6 types of segment registers

CS = Code segment

DS = Data segment

SS = Stack segment

ES = Extra segment

FS = } not have

ES = } full form

The text sends data to CS, Data send to DS, stack send the data to SS

When CS, DS, SS are incapable to hold data it send data to ES, FS, GS

- The Text, Data, stack has big in size and segment registers are small in size. So send the data to segment registers it send in small part. So data is not going waste.

Ig :- Instruction queue

In Ig has to attach the IP (Instruction pointer)

- IP does the work it tells ~~the~~ where the data to be send it gives the instruction.
- If the mathematical operation are performed or Arithmetic operation are performed then the data is send to ALU (Arithmetic Logical unit)
- If non-mathematical data operation are perform the data is send to Control Unit (CU)

• Flag is a register

- It have only 2 states - on, off.
- Flag tells processor to how to perform operations on data
- It is profitable for processor.

* RCU :-

Register control unit

- RCU have the general purpose registers i.e.

AH	AL
BH	BL
CH	CL
DH	DL

there are 4 types of general purpose registers

- A → Auxiliary Register
- B → Base Register
- C → Counter Register
- D → Data Register

this registers are written as

$$EAX = AH + AL$$

$$EBX = BH + BL$$

$$ECX = CH + CL$$

$$EDX = DH + DL$$

↓
higher

→ lower

- ~~Register~~ General purpose register are counted in bits
- If the size of general purpose register is high then the computer speed is also high

* pointer register

- i) Stack pointer (SP)
- ii) Base pointer (BP)
- iii) Instructions pointer (IP)

* index register

- i) Source index (SI)
- ii) Destination index (DI)