

computer Architecture Assignment : 1

Q) What is Toolchain? Elaborate working of every component of Toolchain i.e Editor, preprocessor, compiler, Assembler, Linker, Loader.

→

In C-program has the two phase i.e. compilation phase and running phase and the code travel the time of execution in Editor to preprocessor to compiler to Assembler to Linker to Loader i.e. the Toolchain.

* Editor :-

Editor is a tool w.r.t. we have write something or edit something.

In different operating system has different editors like,

Windows O.S. has,

Notepad

Notepad ++

Microsoft Word

Linux has O.S. has
g edit
k write

Mac O.S. has

pages

sublime

x-code

text edit

* preprocessor :-

Preprocessor is a tool which takes

Input demo.c file from editor and which is gives demo.i file as a output intermediate code.

preprocessor has to do 4 types of works.

- 1) Header file substitution
- 2) Macro Expansion
- 3) conditional compilation
- 4) Line control

→ Header file substitution

→ only handle or process to start the line with # symbol.

There is # called as preprocessor directives.

→ Macro expansion.

→ In # line, # define After the # define to come the function is called macro.

ex `#include <stdio.h>`
`#define MAX 10`

 └→ Macro

→ When program come to preprocessor that time program will be expanded. because in that program added the header files declaration.

* compiler :-

compiler is a tool which takes the input demo.i file from preprocessor and which is gives the demo.asm file as a output.

 └→ assembly.

if program run in Linux O.S. so compiler give a demo.s file as a output.

assembly code is a processor dependent or machine dependent or Architecture dependent

In assembly code there are some mnemonix like PUSH, ADD, MOV, RETN.

* Assembler :-

Assembler is a tool which takes the input demo.asm file from compiler and which is gives demo.obj file as a output.

object

if program run in Linux O.S. so Assembler gives demo.o file as a output.

object

* Linker :-

Linker is a tool which take the demo.obj file as input and which is gives demo.exe file as output.

executable

Linker has work, to take demo.obj file and add other.obj file and gives the demo.exe file as output.

demo.obj + other.obj → object

In this other.obj file have to demo.exe add the definitions. Which is in a binary form.

* Loader

→ Loader is a tool which takes the ~~demo~~ Hello.exe file in hard disk and to bring it in RAM.

Loader has work to take a test to the process which is to bring in RAM.

Loader has ask 4 questions

- 1) What is your magic number?
- 2) What is address of entry point function?
- 3) Which type of executable you are?
- 4) What is your time stamp?

2) What are the different task of operating system?

→ operating system donot work self.

In operating system has 5 man to work.

- 1) process Manager
- 2) File Manager
- 3) Memory Manager
- 4) CPU Scheduler
- 5) Hardware Abstractor.

All process related work handle the process manager.

All file related work handle the file manager

All memory related work handle Memory manager

All CPU related work handle the CPU Scheduler.

All Hardware related work handle the Hardware Abstraction.

Each and every operating system has two types of executable file.

1) self executable

2) dependent executable

To execute the self which do not help to other.

To execute the with the help of the other.

means to depend the other file to execute.

3) What primary Header contains? explain in details?

→ In Hello.exe file divided into 4 parts.
PH, Text, Data, Symbol Table.

PH - primary Header.

In PH have to store the 4 things

i.e. 1) ~~Memory~~ Magic number

2) address entry point function

3) type of executable

4) Time stamp.

→ What is your magic number?

→ To check the number which is same

to the operating system.

Ex

suppose, Window operating system magic number is 100 so they can check with primary header which is same or not. if magic number is same then, they can pass in first question.

2) What is address of entry point function?
→ In C, Java execution always start with main function.

means

our code is 100 line so do not check the line by line. They will direct check the main function to start the execution.

3) What type of executable you are?

→ To check the file form is executable which is executable or file i.e
i) self executable or
ii) dependent executable

4) What is your time stamp?

→ When you are made or created.
Time to check the time when file is created.

4) Explain in details Text, Data, Symbol Table and Stack section.

→ Text :-

Text :- It will be store the binary instruction or code. And this data has unit is bits.

→ Data :-

In this data section, it can be store the global variable.

Data section

BSS :-

Block starting with symbol.

In this, if global variable is not initialize so to allocated or to get a memory has in BSS.

Non-BSS :-

Block starting with value.

In this, if global variable is initialize so to allocated or to get a memory has in Non-BSS.

3) Symbol Table :-

Which contain information about variable and symbol.

symbol table which is a meta data

Symbol Table do not allocated the memory which is shows only information.

Symbol Table has 8 entries are there available.

4) Stack section :-
In code Local variable are allocated the memory has to get the stack section.

In Text function, if fun at the time of function call that time function has frame and this frame has allocated the memory to get in stack section.

→ Explain the working of ECU, IQ, RCU?

→ * ECU :-

Execution control unit

It has 6 types of segment Register
code segment = CS

Data segment = DS

Stack segment = SS

Extra segment = ES

~~It has~~ { = FS

no full form } = GS

The Text send data to es , Data send

To DS, stack send the data SS.

Text, Data, Stack has big in size and Segment Register small in size.
Hence, To send the data to segment Register has to send in little-little part.

If CS, DS, SS has to less to hold the data so ES, FS, GS that time use to hold the data.

ex

Suppose, Text, Data, Stack holds 16 bits data and send the data to CS, DS, SS and it's size only 10 bits. So data will be wasted, that time data sent in little-little form.

means

16 bits, after execution 10 send 10 bits after execution 10 bits. That types of data is sends.

* IQ :- Instruction Queue.

In IQ has to attach the IP (Instruction pointer). and IP is work to get the data and to take the decision where is data send.

if mathematical operation is performed or Arithmatical operation is performed so data or process will be send to the ALU (Arithmetic Logical unit).

if Non-mathematical operation perform to be instruction so data or process will be send to CU (control unit).

FLAG is a Register.

FLAG has two states
on or off.

FLAG Register has to most benefit of has to processor.

processor has decision maker.

example of FLAG

register

if suppose, Railway has to run in the road and exactly that time to get the driver a red signal so drive has stop the railway. driver do not tell why give me the red signal my railway donot stop here, because red signal mean stop the train in any condition. means driver has write right to take the decision on the spot.

same as like processor has to take the decision if FLAG on or off.

Hence,

the processor called decision maker

* RCU :- Register control unit.

RCU has to General purpose Register and i.e.

AH AL
BH BL
CH CL
DH DL

4 types of General purpose Registers.

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

and it has to write in this manner

$$EAX = AH + AL$$

$$EBX = BH + BL$$

$$ECX = CH + CL$$

$$EDX = DH + DL$$

↓

H = Higher

↓

L = Lower

General purpose Register has count in bits.

General purpose Register has depends the speed of your computer

pointer Register.

(SP) Stack pointer

(BP) Base pointer

Index Register

(SI) Source Index

(DI) Destination Index.