PAGE No.
DATE / /

Assignment - 1

Computer Fundamentals & Architecture

1) what is computer?

→ - Computer is an electronic device

- which is use to perform simple as well as complex operations to solve simple as well as complex problem with the help of input and output devices as well as central process unit (CPU)

2) what are different components of computer

→ Hardware is thing which is we can touch and fell its hardware

- following are the different components of computer

1) Microprocessor

2) Math co-processor

3) CPU

4) Storage device

5] Input device

6] output device

7] motherboard

8] System Bus

3] what is ~~devi~~ storage device and types of storage devices?

→ Storage device

- Storage responsible for storage the data.
- It can used for storing information for temporary and permanent storage.
- There are 2 types of Storage device

1) primary storage device

2) Secondary storage device

PAGE No.
DATE / /

1) Primary storage device

- Generally storage device Accesssable to microprocessor directly are called primary storage device.

e.g:- RAM

- It is primary electric device which is used to store data temperary
- Every program gets loaded into RAM for execution purpose

Rom

- Rom is electrical primary storage device
- Every Rom becoming a Rom is considered as RAM

2) Secondary storage device

- This storage device are used to store the data for permenantly
- Before the data get loaded into secondary it initial store in the primary.

- It is compulsory and mendatory shift the data from primary to secondary permentatly
- when data is not in RAM we cannot shift into secondary

Hard drive

- It is secondary
- used to store the data using mangetism
- when we create the file & save that file it gets into secondary storage

4] What is mean by Toolchain or x86 Toolchain?

→ Toolchain

Toolchain is considered as set of software (tools) which are use to convert human understable program into Machine understanable program (binary)

- using toolchain convert human understand to machine understandable code

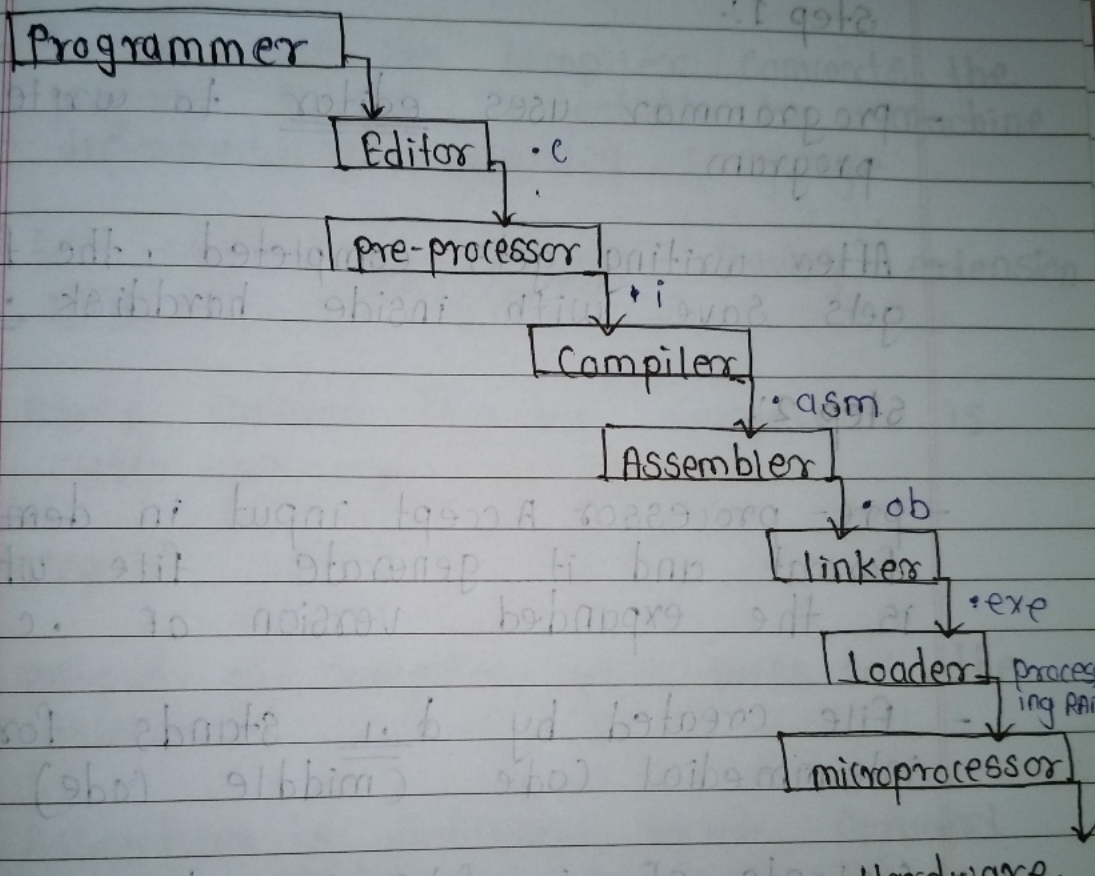


fig: Toolchain X86

PAGE No.
DATE / /

- Above fig shows the toolchain ~~of~~ of x86

Step 1:-

- programmar uses editor to write the program
- After writing gets completed, the file gets save with inside harddisk .c

Step 2:-

- pre-processor Accept input in ~~dem~~ .c format and it generate file which is the expanded version of .c file
- File created by .i stands for intermmedial code (middle code)
- contents of .i file are human readable and understable

Step 3:-

- output of pre-processor gets provided as input to the compiler

- Compiler is software which convert program from one language to another
- In our case the compiler converts the program human undestable to machine dependant i.e Assembly language
- file created by Compiler having extension .asm
- newly created file by compiler is .asm or .s

Step 4:

- output of compiler gets pass to the Assembler
- Assembler is software which convert program machine dependant into machine undestable format
- output of Assembly having file extension .obj
- .obj contain the code in binary format but it is not directly executable.

Step 5:

- linker is responsible to link .obj file generate by the Assembler and it dependant .obj file
- linker generate output with extension .exe
- In our case .exe is output of linker

Step 6:

- .exe generate by linker currently stored inside harddisk
- To execute any Application or program is has to be present or located inside RAM
- After loading the .exe file into RAM it get considered as process and it get executed with the help of operating System.

PAGE No.
DATE / /

5] What is CPU register? what are types of it? and explain use of each CPU register.

→ CPU register

- CPU registers are small fast storage areas in a computer's central processing (CPU) that temporarily hold data & instructions while the computer is working

- following are multiple CPU register

AH, AL = (AL) Arithmatic register

BH, BL = (BL) Base register

CH, CL = (CL) Count register

DH, DL = (DL) Data register

SP = Stack pointer

SI = Source index

DI = destination index

PAGE No.
DATE / /

Arithmatic register (AL)

- AH and AL registers are 8 bit registers that are paired together to form the 16-bit AL register.
- AH is the high byte and AL is the low byte.
- AL and AH are ~~8 bit~~ the 8 bit "char" size registers.

Base register (BL)

- BL is known as the base register it could be used in indexed addressing.
- It can be also point to a byte in memory or the beginning of a memory array.

Count register (~~CH~~) (CL)

CL is known as the data register ~~it~~ is ~~also used in input~~ CH, CL store the loop count in iterative operations

PAGE No.
DATE / /

Data register (DL)

- DL is known as the data register it is also used in input / output operations it is also used with DH and DL registers for multiply and divide operations involving large values

Stack pointer (SP)

- A stack pointer is a register in a processor that stores the memory address of the most recently added item to a stack
- It's also known as the extended stack pointer.

Source index (SI)

The source index register (SI) is a 16 bit general purpose register in a computer's CPU that's used to identify memory addresses in the data segment that the data segment register (DS) is addressing

Destination index (DI)

- The Destination index (DI) register is a 16-bit general purpose register in the microprocessor it is used as a memory pointer to store the

6] what is operating system & different tasks of O.S?

→

operating system

- operating system is system software that manages computer hardware & software resources and provides common services for computer program

- operating system (OS) is the program that, after being initially loaded into the computer by a boot program, manages all of the other application programs in a computer

- following are the tasks of the operating system.

1] file management

2] process management

3] memory management

4] CPU scheduling

5] Hardware abstraction

7] Explain working of each tool from toolchain i.e Editor, preprocessor, Compiler, Assembler, linker, Loader

→

Editor:

- programmar uses the editor to write the program.

- After writing gets completed the file gets save inside harddisk in .c format.

Pre-processor

- pre processor Accepts i/p in .c format it generate file which is the expanded version of .c file

- file created by .i stands for intermediate code in (middle)
- file generated by pre-processor is human readable & understable

Compiler

- ~~outp~~ Compiler Accepts the input from the pre-processor in .i format
- Compiler is software which convert program from one language to another language
- Compiler convert the program human understable from to machine dependant i.e Assembly language
- The output of the compiler generated file is .asm or .s

Assembler

- Assembler is software which convert program machine dependand into machine understable format

output of Assembly having file extension .obj

- .obj file which generated by Assembler is in binary format but not executable

linker

- linker is important tool in the toolchain it is responsible to link .obj file generated by the Assembler & it machine dependant obj file.

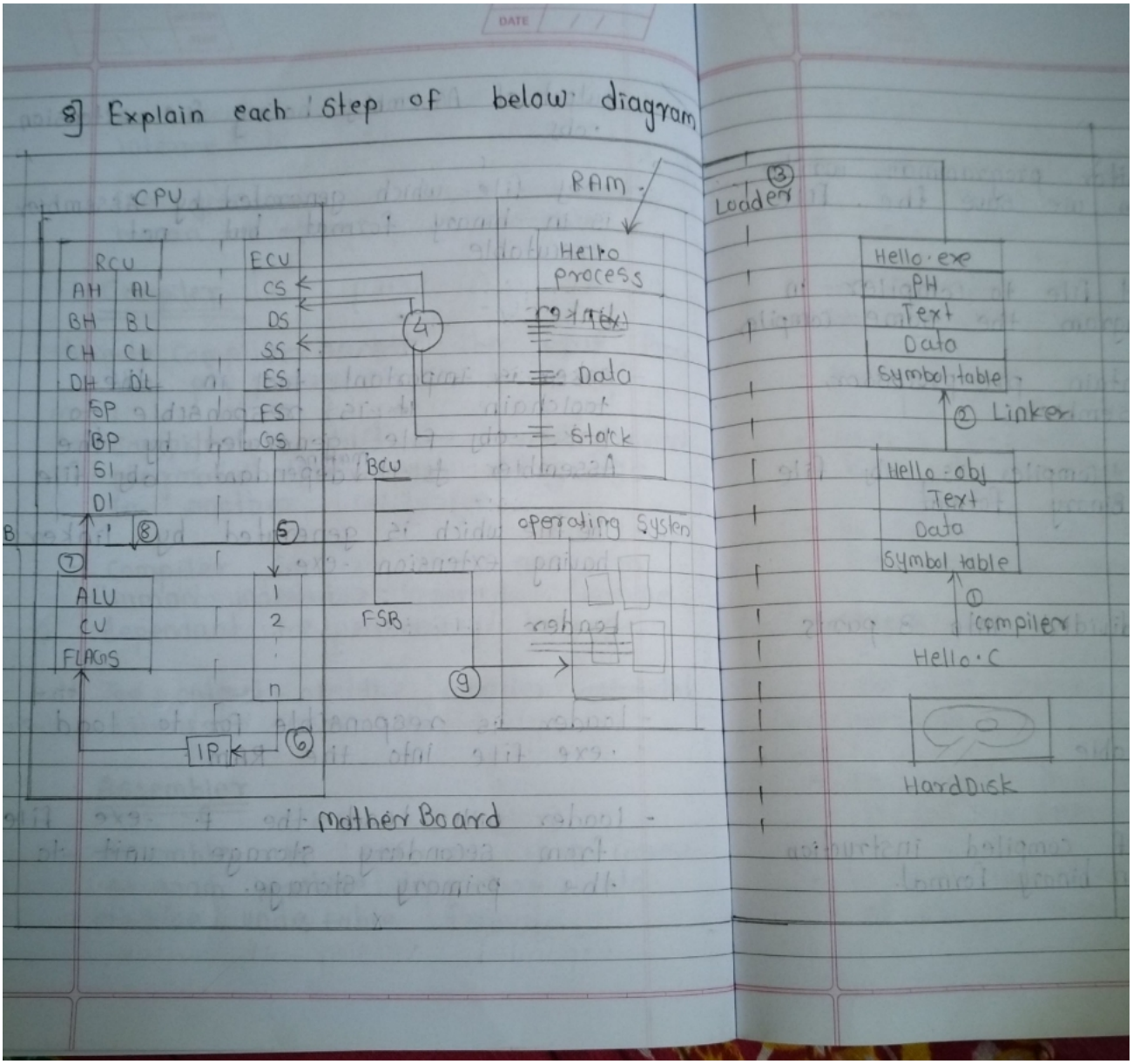
- The file which is generated by linker having extension .exe.

Loader

- Loader is responsible for to load .exe file into the RAM

- loader will load the .exe file from secondary storage unit to the primary storage.

8] Explain each step of below diagram



PAGE No.
DATE / /

Step 1:-

- By using editor programmar write the program we save the file as hello.c

- we pass that file to compiler in our ~~proo~~ diagram the name compiler

- internally contain pre-processor, compiler, Assembler

- The output of compiler is .obj file which is in Binary format

Step 2:

- .obj file is divided into 3 parts

1) Text
2) Data
3) symbol table

1) Text

Content of compiled instruction of a program in binary formal.

2) Data:

Data section contain memory for global variable use in program

3) Symbol table

It's is table which contain the information about symbol character, special symbol which is used in program

- Now we pass .obj file to the linker
- linker is responsible to link our .obj file with it dependant
- the linker create .exe file & which considered as executable file.

Step 3:-

- .exe file created by linker stored in harddisk
- to execute that file it should be loaded in RAM.
- to load that file into RAM the loader of OS is responsible.

- Inside the .exe file a PH (Physical memory header) is added
- PH contain information about .exe file

step 4:

- when the loader load that file into RAM it is considered as process
- process into RAM is divided into 3 parts
  1) Text
  2) Data
  3) stack
- The stack section contain information about the function which are written inside our program
- for each function there is stack frame.
- RAM is not responsible for execute the process due to which we pass that process to microprocessor
- inside the microprocessor there is no sufficient memory like RAM due to which direct loading text, data, stack

is not pratically possible.

- So that each section is divided into multiple segments.
- text segment gets copied into ~~to~~ CS (code segment).
- Data gets copied into DS (Data segment)
- Stack gets copied into SS (stack segment)
- If the ES is also full FS and GS are used.
- CFS & GS has no any full form

Step 5:-

- Now the data from segments gets copied into instruction queue
- ~~for~~ from that instruction queue single instruction gets fetch at a time & forward that instruction at the time

PAGE No.
DATE / /

Step 6:

- Now the instruction gets forward to ALU if instruction related the Arithmatic operation.
otherwise
The instruction gets forwaded to Control unit.

Step 7:-

- In the ALU unit flags indicate the internal instructions of instructions gets store forwad to CPU
- Now instruction gets stored in CPU registers.
- There are multiple CPU registers depends on their requirement.

Step 8:-

- Inside ALU CPU register execution of instruction gets performed.

Step 8:

- The instructions in the CPU register & the output of instruction gets forward to IDB

Step 9:

- After getting the output of the instructions are forwaded to the operating system through Bus control unit & the front side Bus
- After the output getting operating will display the result on console.

9] what are the contents of physical Header?

→

Physical Header

- By using the physical header operating system will undestand to load the loader or not

- physical header store the address of main function.
- following are the contents of physical header

1) Text

2) Data

3) symbol table

10] what is meant by Text, Data and Stack Section?

→ **Text:**

- The program in the RAM is divided into Text, Data, Section
- In the text there is the binary format about the executable file is stored

Data:

Data section Contain memory for global variable in the program

PAGE No.
DATE / /

Stack Section

The stack section contains information about the function which are written inside your program.

- for each function there is stack frame.