**Date: 20-10-2022 Lecture # 2**

* In CLI, we must know the path of a directory if we want to open it.
* **$** 🡪 represents regular user.
* **#** 🡪 represents root/ administrator.
* Whatever we write after **“$”** sign, is a command.
* Some commands need arguments like *mkdir, cd*, etc. While, some commands do not need arguments like *ls, whoami*, etc.
* $pwd is command which tells the present working directory.
* There are two ways to access a resource:

1. **Absolute Path:**

This path always start from the (/) root.

1. **Relative Path:**

This path is related to only the present working directory.

It does not start from /.

* To go back on parent directory, use **‘. .’** before the command.
* **USERS:**
* Users have accounts on the system
* Users write programs, then execute programs
* Different users may execute same program
* **DIFFERENTIATE BETWEEN**

**PROGRAM VS. PROCESS**

|  |  |
| --- | --- |
| **PROGRAM** | **PROCESS** |
| * A program by itself is not a Process. * Programs are executable files which are created when you run source files on computer. | * A program in execution is a process. * Instance of a program running on a machine * Moving program/executable file from hard disk to main memory, is called process. |

**Conversion of Source Files into Executable Files (PROGRAMS): (COMPILATION PROCESS)**

* This conversion occurs in four stages:

1. Preprocessing:

In this phase,

* Comments in files are removed.
* Header files are included.
* Macros expanded (defines, etc.)
* It converts **‘.c’** file into **‘.i’** file. (It converts hello.c to hello.i)
* We use gcc E to execute this

1. Compiling:

In this phase,

* Syntax of **‘.i’** file is checked.
* Converts code into assembly language.
* It converts **‘.i’** file into **‘.s’** file. (i.e it converts hello.i to hello.s)
* We use gcc s to execute this

1. Assembling:

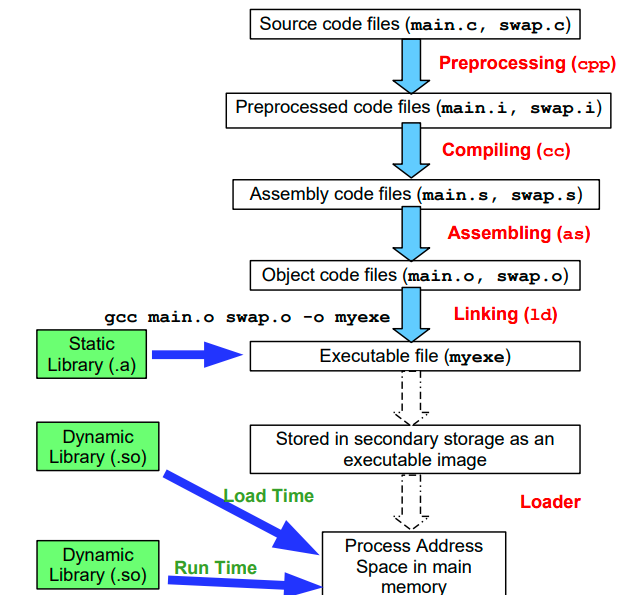
In this phase,

* Machine code is generated (binary code).
* It creates executable file but it cannot be run.
* It converts **‘.s’** file into **‘.o’** file. (i.e it converts hello.s to hello.o)
* We use gcc c to execute this

1. Linking:

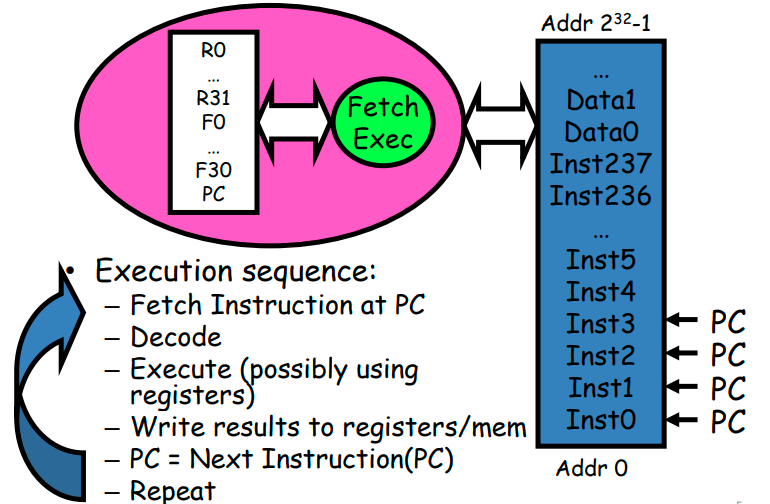
In this phase,

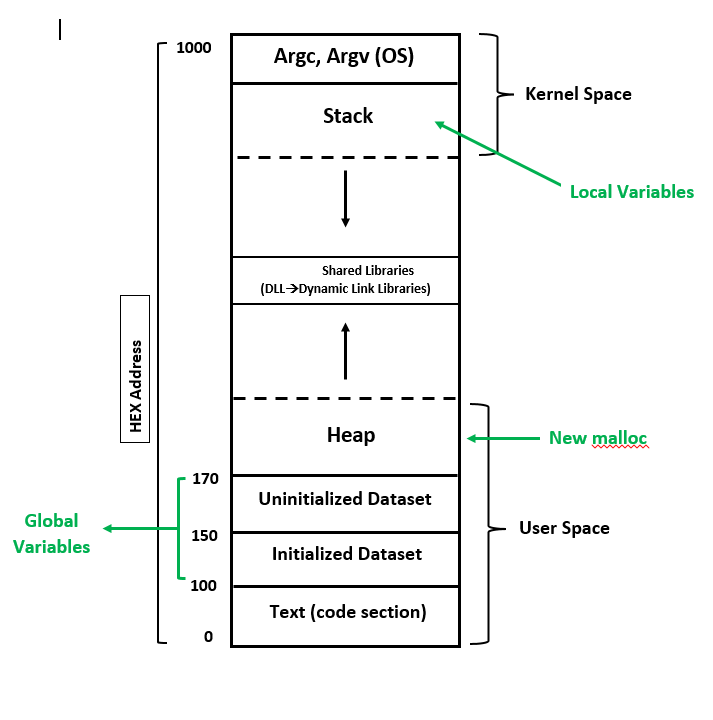
* All libraries are linked with code.
* It creates executable files which are called as ***‘programs’.***
* This program can only run on your OS as it present in your hard disk.
* Here a file created which has extension a.out .
* We use gcc to execute this

****

**CPU:**

* CPU contains some registers and a PC.
* PC is a program counter. It contains addresses of process from where information is fetched.



**Process A****ddress Space**

* ***readelf*** and ***objdump*** Commands:

These commands are used to see everything in a program/executable file.

* **/proc** Directory:

It is used to check which command do what.

It has details list of all the commands

* Two things are passed to a program:

1. **Command Line Argument:**

* The arguments written after a command.
* Program counts its own name in arguments.
* **Example:**

Cat (command) f1.txt (argument)

1. **Environment Variables:**

* Every program requires an environment to run/execute.
* If we don’t give any environment, then OS gives/uses its own environment (default) to program.
* When we run a program, we need to provide better/best environment to it.