Program Vs Process

→ Program.c ←

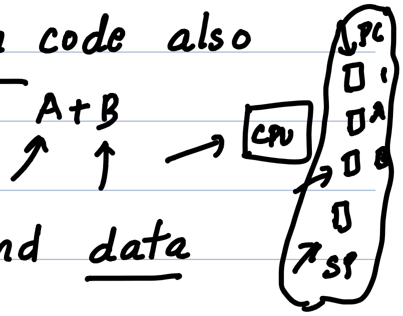
→ gcc program.c → a.out

→ ./a.out

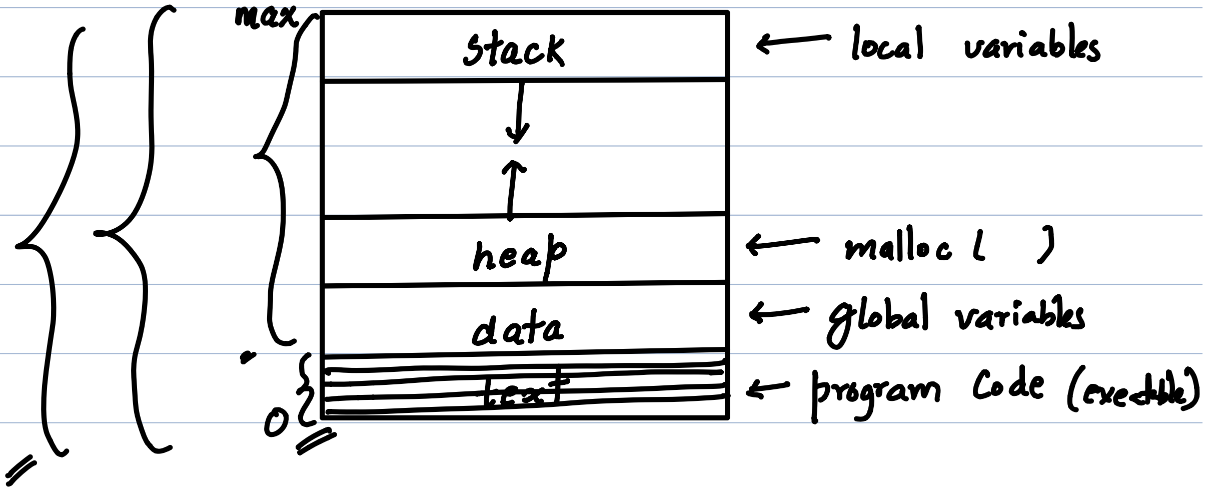
- A program is just an executable file which is stored on the hard disk or secondary memory.

- When we execute this file either by double clicking or from the terminal, the program becomes a process.

- A process in addition to the program code also consists of other attributes :



  - Memory — stack, heap, text and data



  - Program Counter : The address of the next instruction to be executed on the CPU.

  - Process state : New, ready, running, waiting etc.

- CPU registers : The values of CPU's registers must be stored when an interrupt occurs.

- dist of open files : List of files the process has opened for reading and writing.

- Process priority - used in process scheduling

- Process id : Unique identification number of each process

Init ①

- Parent process id : Who is the parent of the process ?

These all attributes/information of a process is stored in a data structure called Process Control Block (PCB).

- So, when we double-click an executable file or execute it from the command line, a process control block gets created for it.

pcB

```
struct task_struct {
    pid_t pid;  ←
    long state;  ←
    int priority;  ←
    struct task_struct *parent;  ←
```

```
        struct files_struct *files; ←
        struct mm_struct *mm; ←
        long counter; ←
        - - -
        - - -
    };
```

PCB

Process ID ←
Pointer to Parent ←
List of children ←
Process state ←
Pointer to address space info ←
Program Counter ←
stack pointer ←
other register values ←
open file list ←
Scheduling priority ←
Parent ID ←
- -

- Always remember that a program is a passive entity. A process is an active entity. A program is just the text section of the process. //