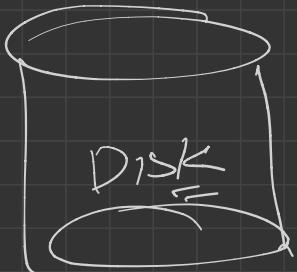
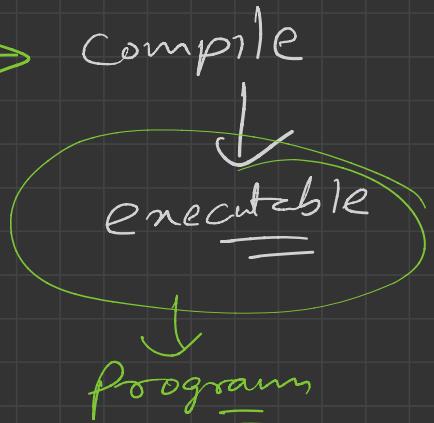



Lec - 9

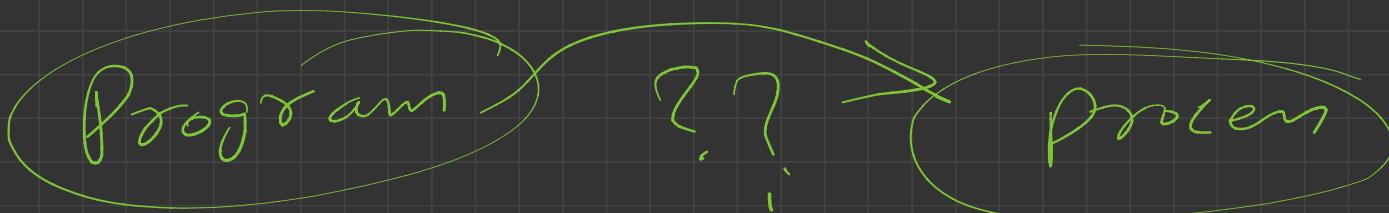
Process? What=



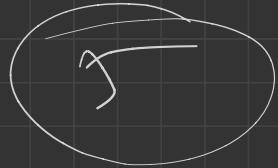
Program e.g. TikTok $\xrightarrow{\quad}$ OS \rightarrow Program

\rightarrow program under execution $\xrightarrow{\quad}$ process
 $\xrightarrow{\quad}$ process

Why? process \hookrightarrow User - Work. \rightarrow way



Now OS creates a process.



- ① Load the program & static data to memory
↓
Used for initialization

char *name = "Lakshay";

int a = 0/1;

②

Allocate Routine stack.

→ Part of memory used for
local variable, fⁿ argument
f return value
=

③

Allocate. heap :-

→ Part of memory used for
dynamic allocation
=

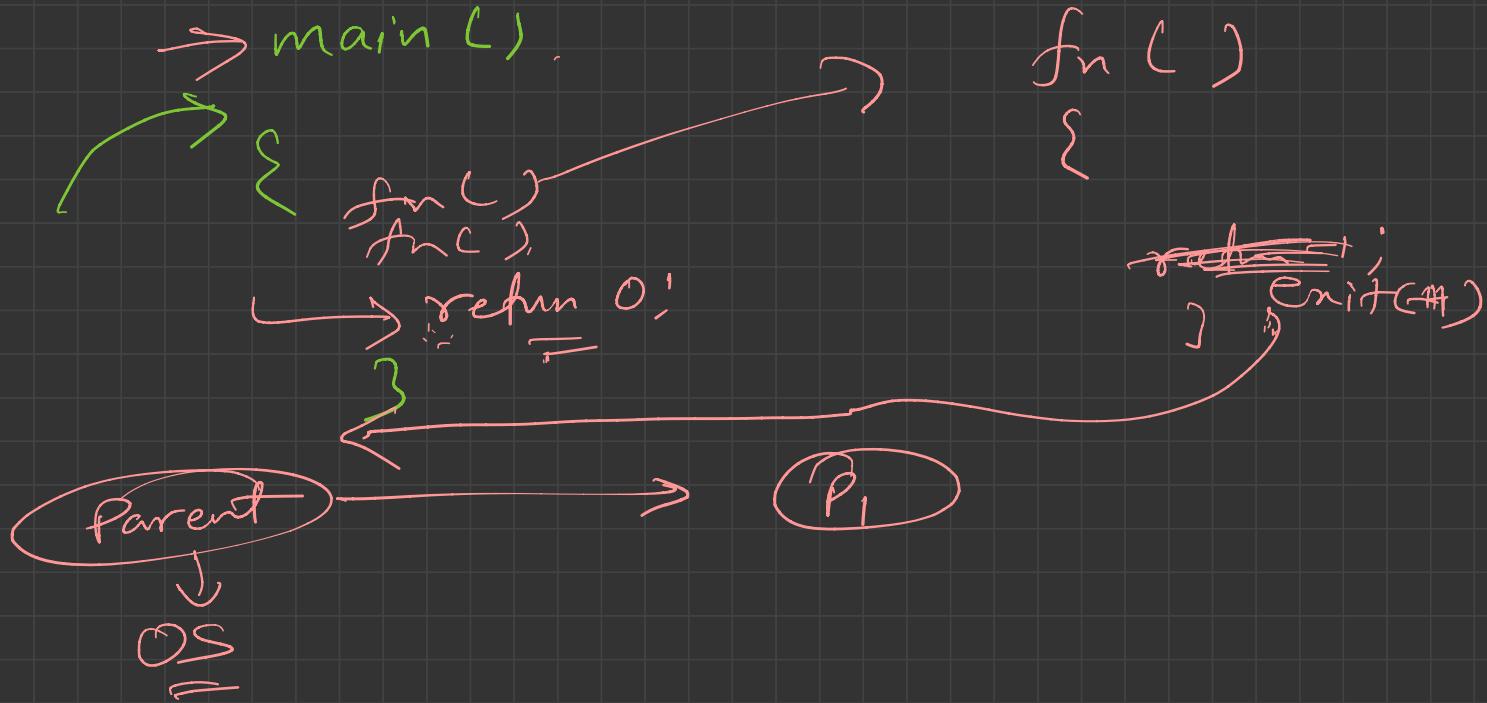
10 tasks -

Unin ← i/p → handle
o/p → o/p →
error → handle.

fprintf(stderr, "hang");

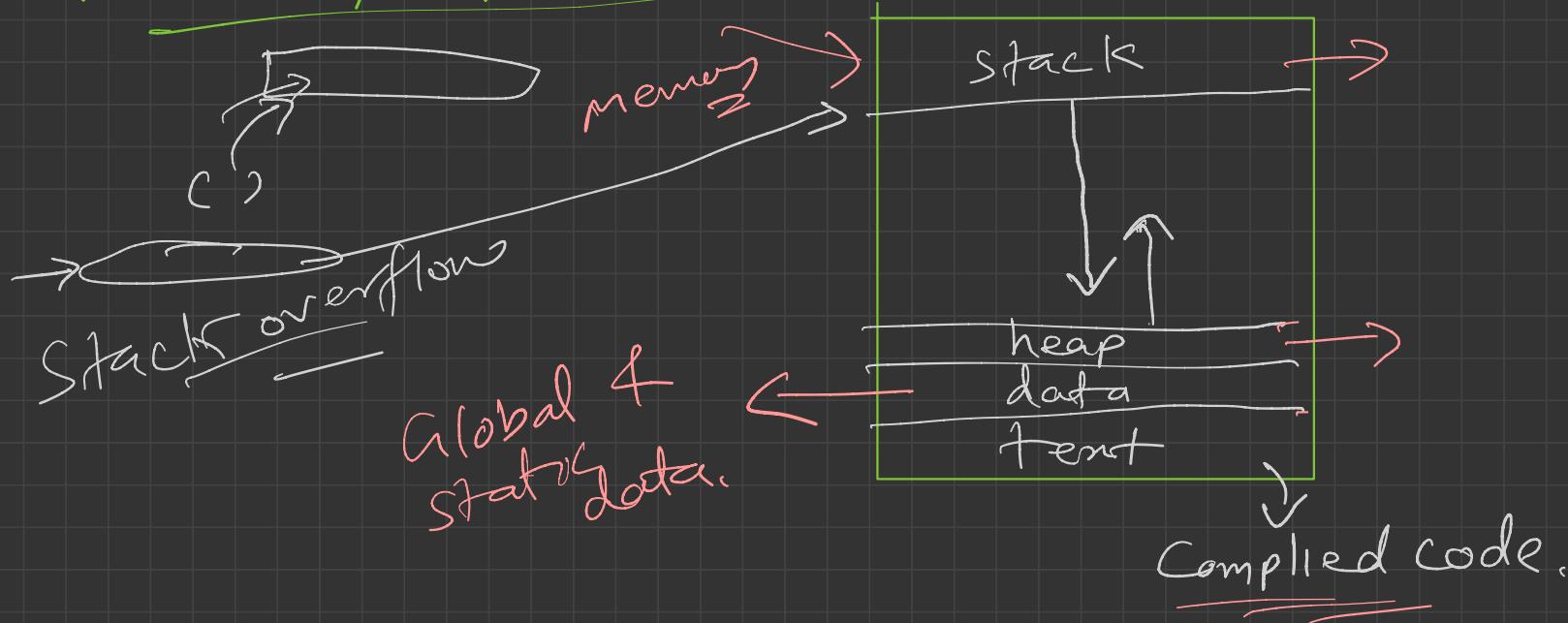
5

OS handoff control to main()



Process → memory

*Arch. of Process :-



- ① Stack overflow — Bone care set.
- ② Out of memory — deallocate unnecessary object.

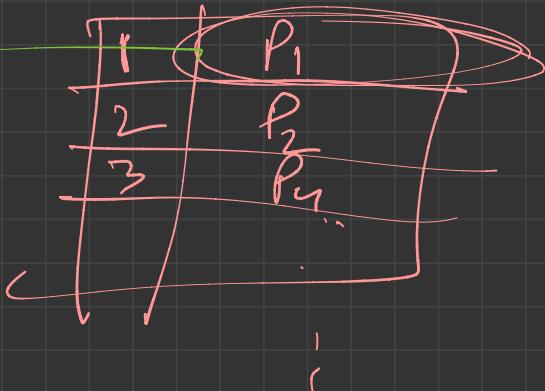
Attributes of process :-

$P_1 \ P_2 \ P_3 \ P_4 \rightarrow$

Process Task

PCB

Process Control Block



SP → stack pointer

BP → Base pointer

CR →

CPU

D D D

Process ID
Program Counter
process state,
Priority
Registers
open File List
open Devices List

→ unique identifier

→

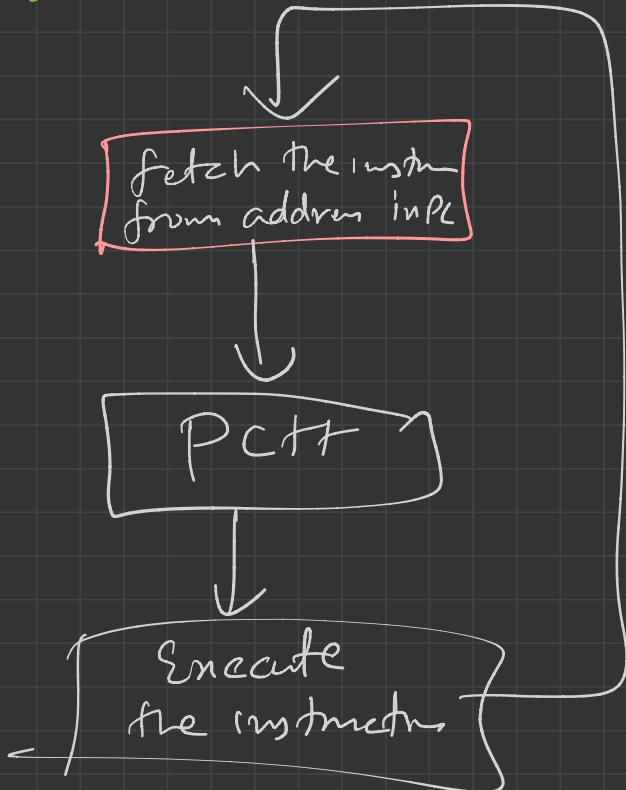
New, wait, RUN.

→ Prior.

Saves register
CPU.

PC B

Program Counter



1 2 ADD =

{

①

②

③

④

⑤

}

}

}

}

}

3