Operating Systems (CS3000)

Lecture – 12

(exec(), wait() System Call)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING,
KANCHEEPURAM

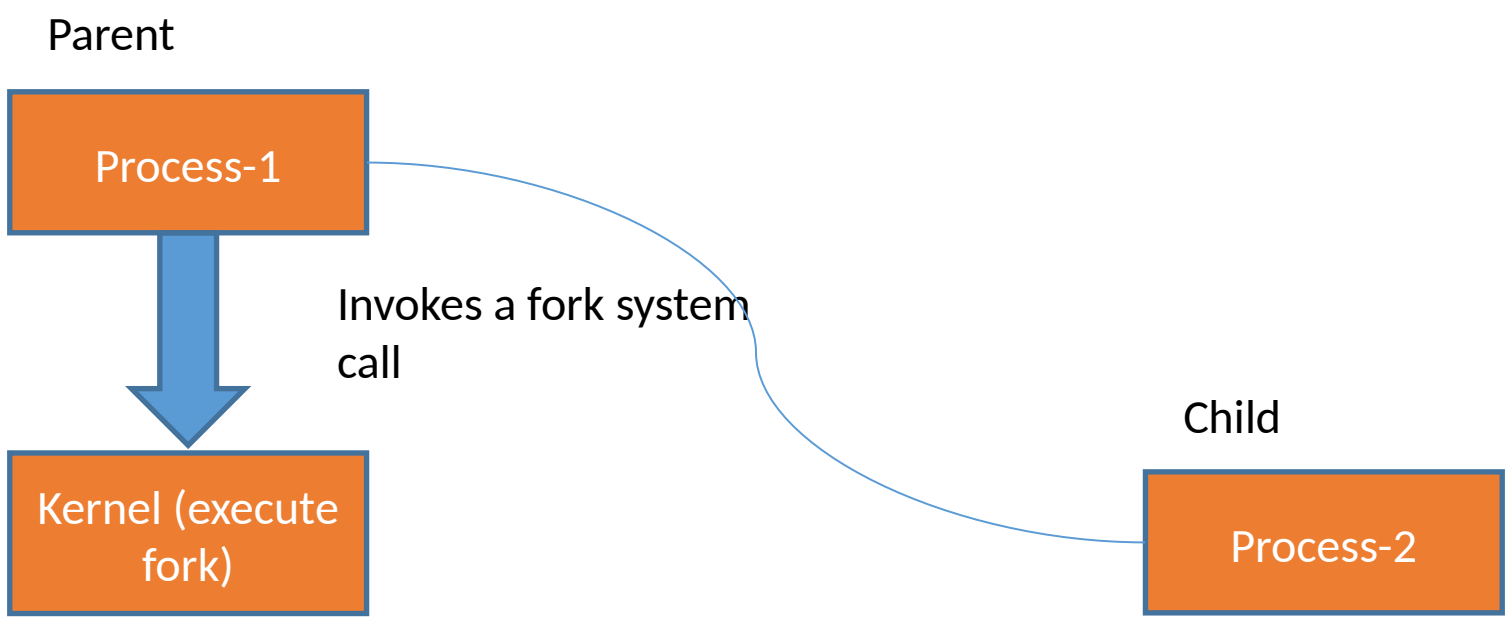
Dr. Jaishree Mayank

Assistant Professor

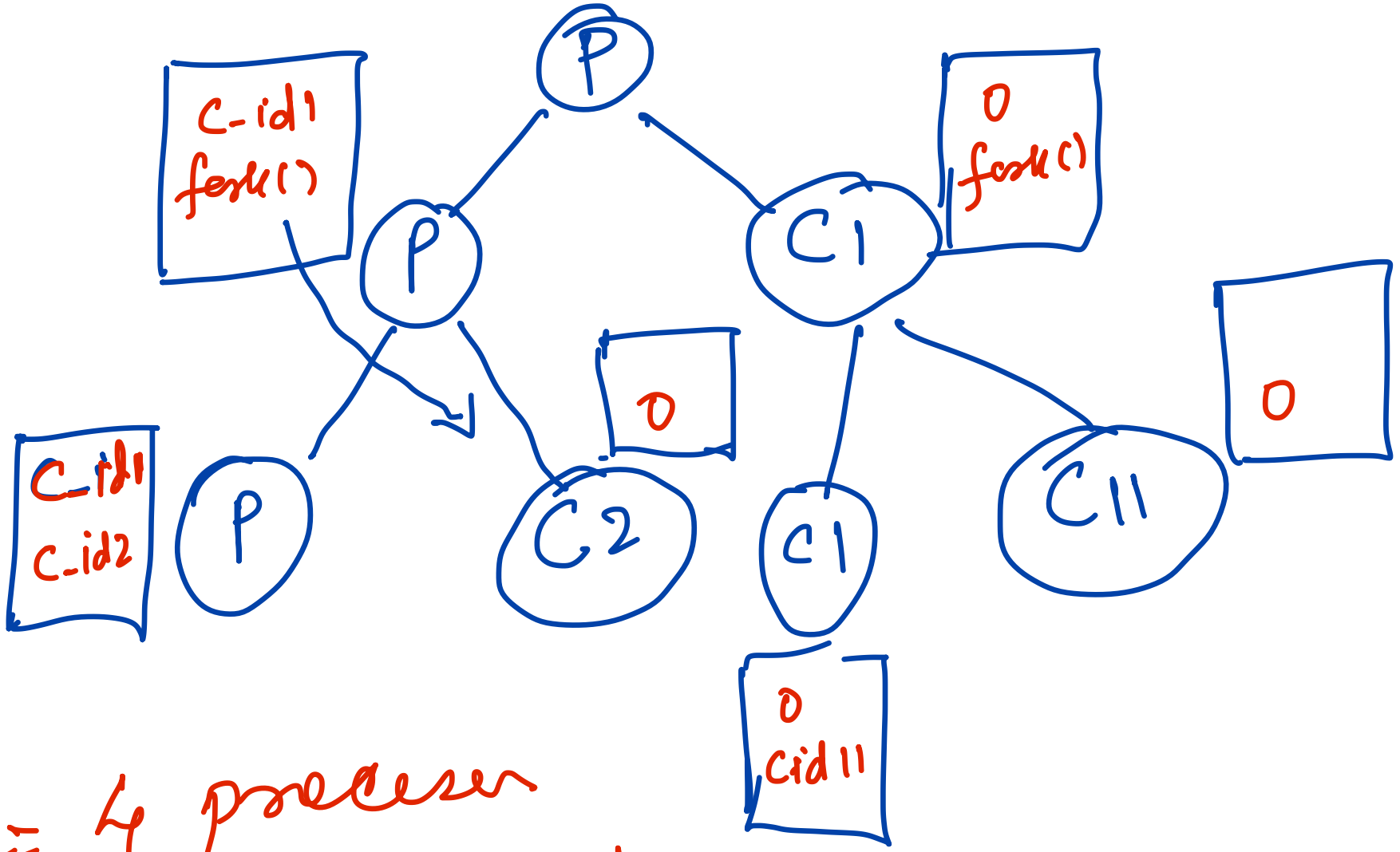
Department of Computer Sc. and Engg.

Creating a Process by Cloning

- fork()
  - Child Process is duplicate of parent process
  - PID ▭ Parent process is Child's PID
  - PID ▭ Child process is 0



```
{
    fork();
    fork();
}
```



Total = 4 processes

1 parent + 3 child

The classic fork( ) Bomb!

- Overall processes in Main Memory as indicated at the leaf level nodes count
- In general n fork class (non conditional) will result in _____processes

```
int main()
{
fork();
fork();
fork();
printf("Magic of fork()\n");
return 0;
}
```

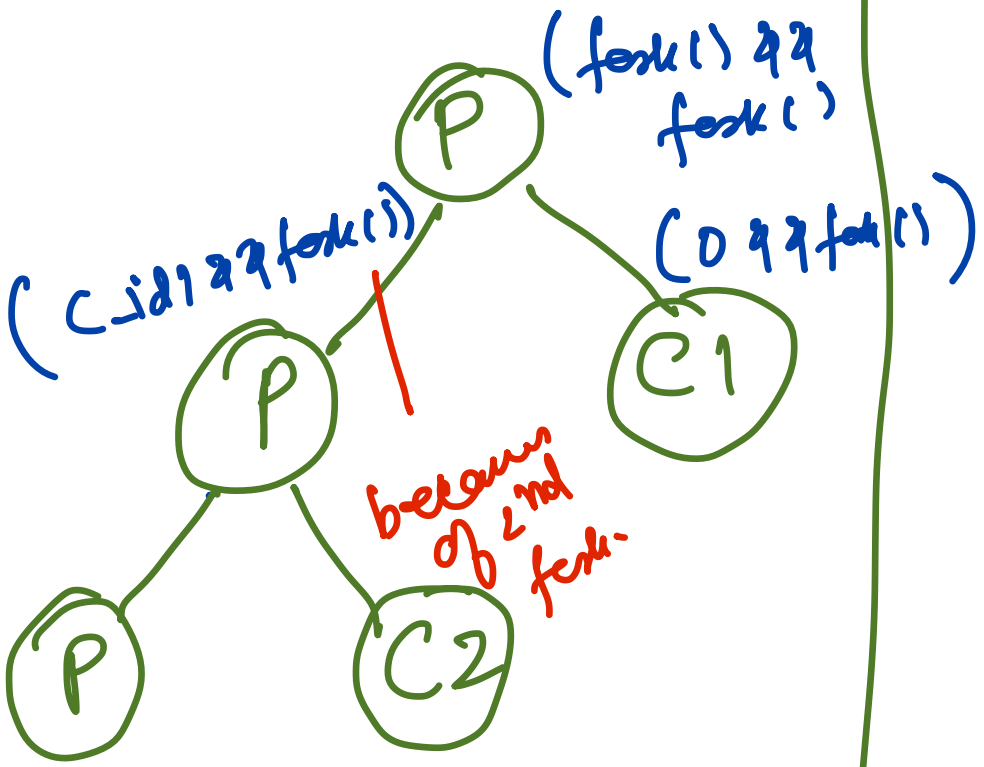
if (fork() && fork())

{

}

else {

}



if (fork() || fork())

{

}

else {

}

Draw Process Tree?

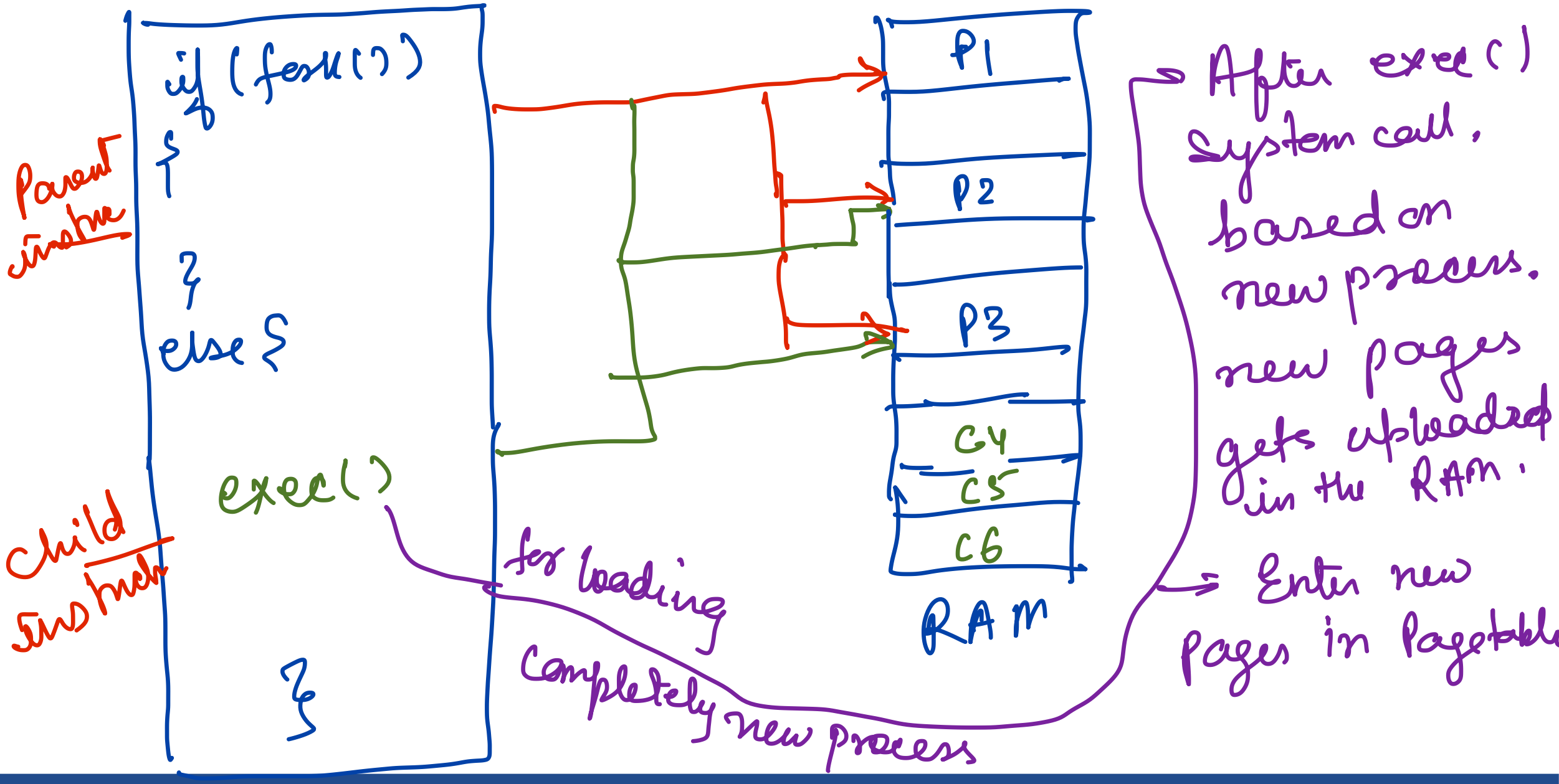
How to create a new definition of process?

- So far fork example we did, child process carried the same image as the parent process.
- Practicality requires child to have new definition.
- Is it possible?
  - Yes

exec( ) system call

- fork() system call is used to create a SEPARATE, DUPLICATE process with **non-shared pages** from which process (parent) it is called.
- Using **Copy-on-Write (COW)**-SEPARATE, DUPLICATE process with Shared pages
  - The new child process will have different PID.

- When exec() system call is invoked from (p1), the program specified in the parameters of exec() will replace the entire process.
  - exec() takes one parameter, which is another program(p2)
  - p1 will be replaced by p2.
  - Replace one process with another process (PID don't change)
    - As we are not creating new process
    - We are replacing an existing one
    - Same PID with different content


if (fork())
{
}
else {
exec()
}
Parent instruction
Child instruction
P1
P2
P3
C4
C5
C6
RAM
for loading completely new process
After exec() system call, based on new process, new pages gets uploaded in the RAM.
Enter new pages in Pagetable

Copy on Write

* Initially parent and child process share the same pages.
* If any of the parent or child wants to update any page, a new copy of that page will be created for that particular page.

wait( ) system call

- called in parent process
- int wait(arg);
- Parent goes to block state
  - Until one of it’s children terminates
  - -1: if no child is executing or exists

- When the child process exits i.e exit(0), it would cause the parent process to wake up
- the wait function returns the child process’s pid

- The parent waits for the child process using this system call.

Why wait( ) system call ?

* Waiting for the completion of child process.
* Sleep(t) system call can also be used but it will not guarantee that once child completes its execution, then only parent will start its execution.

Thank You

Any Questions?