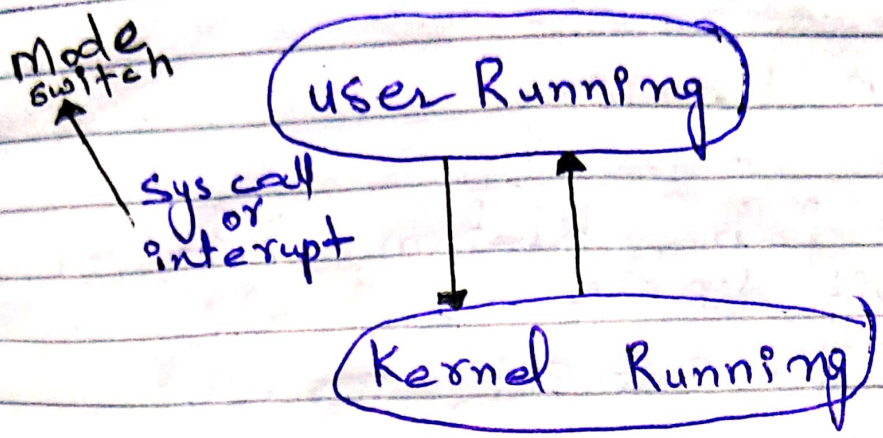
5-lecture Process States, Process Creation, Fork System call.

Ready Queue:-

⇒ All the processes that are ready to run will be in "Ready Queue"

⇒ Har device ki apni apni queue hoti hai.

⇒ Process states.



⇒ Ye jb kernel mode mn gaya bhi. And ye Ausi ek same process ki baat horahi, we can't say that "Kernel apna kaam shuru kardeg".

Created state.

⇒ Jab process pehli dafa ata hai tb seedha to ready mn nhi jaega na.

⇒ Aus waqt wo created state mn ata. (Jab memory Mapping ho rahi hoti hain)

⇒ Jab OS ausa initial resources (Memory wagera) allot kardeta hai, aus waqt hum

kah sakte hain. Ye ready state mein hai

"House keeping"

⇒ Process jab khatam hota hai toh exit ki ek call aati hai. exp:- int g1, exit

⇒ Ye saare ke saare "sys call" hain. inki wajah se hum jate hain kernel mode m

⇒ Phir kernel ussa clean kardeta hai.

⇒ To the point def of (House keeping).

Jitna bhi apke pass resources allocate kiya gaya thai kisi bhi cheez ko (In this case) process. Aur saaro ko free up karne ko House keeping kaha jata hai.

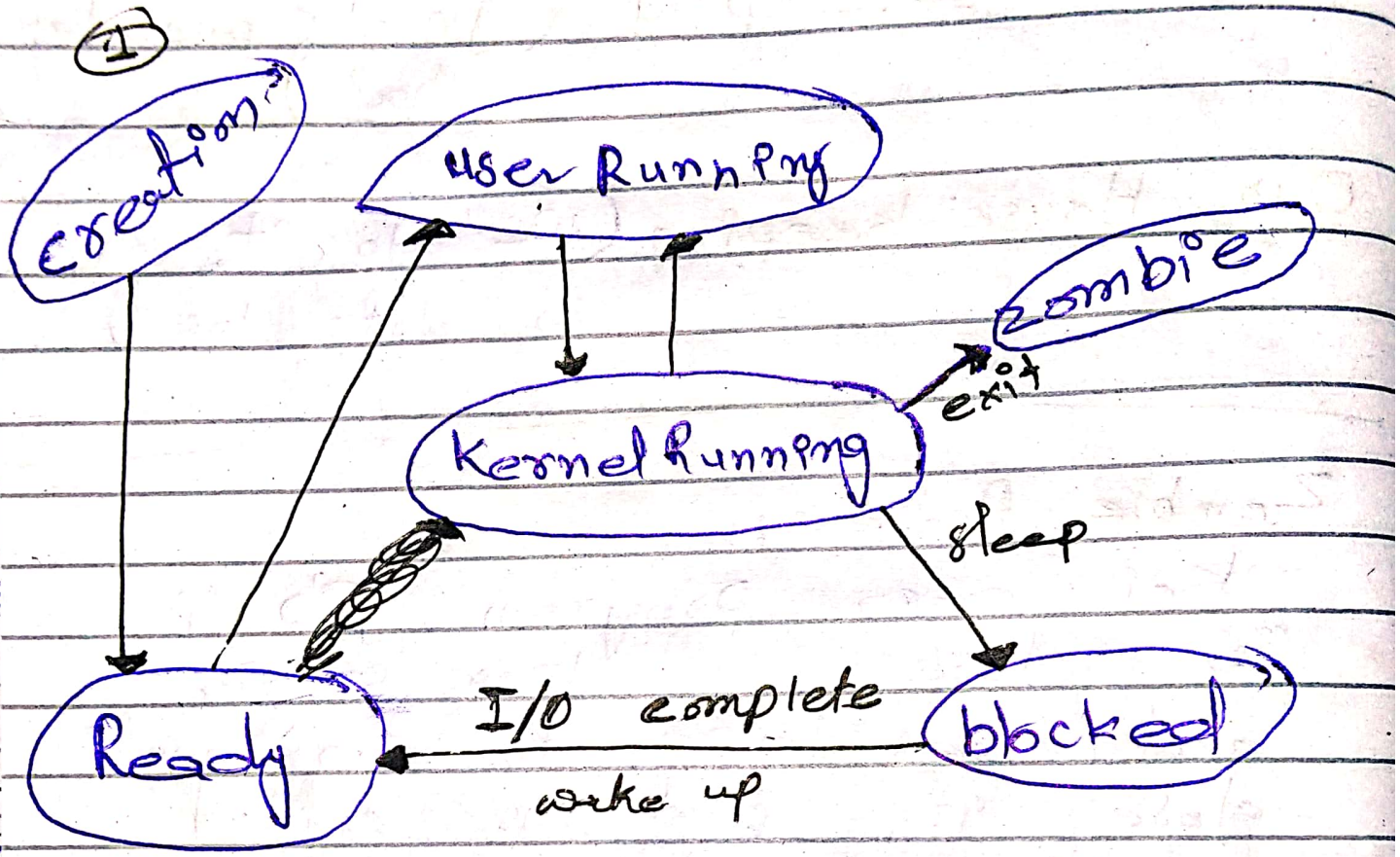
OS → Housekeeping (jhaddy pherdaty hai) → Data loss

Our House keeping (We do it, we'll = no data loss)

Zombie Process:-
Koi aesa program jise OS close na kar sake and oe have to go and close it, that is called zombie process.

Zombie state:-
Whenever the process dies. or exits, it goes into zombie state. Then some house keeping is done, then the process is finished.
tata, see u, bye :)

Bpg Picture



=> Har Process ki Information in Teen cheegon mn store hoti hai

Memory Map CPU PCB

(Process Creation)

⇒ Ek process pehle OS create karta hai, which is init.

⇒ When we click on Application us tym OS process ko create nhi karta. woh kese hota hain. woh hum. Pdhai dekhng chg gahe hai.

Creating a Process under POSIX

⇒ Fork system call.

process execute
= =
= =
= =
Fork();

↓

kernel mode

is process kg duplicate bongay and Abb do processes "Ready state" mn chalegaye.

What happens ??

① Check for available resources.

(Resources hain ke be nhi)

Exp:- ek user 500 process create kar sakta hai

② Allocate a new PCB

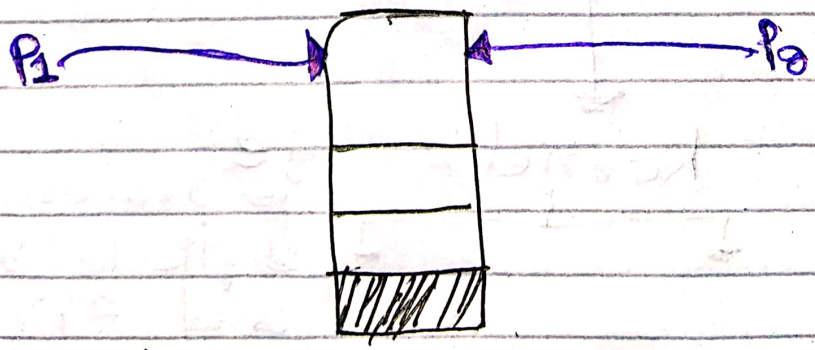
③ Assign a unique PID

④ check process limit user

⑤ Set child state to "created"

⑥ Copy data from parent PCB slot to child (Except "PID")

Abb yahan pe dono ki memory Mapping bhi same hoga kuch is tarha



=> Is it good thing ~~or bad~~ bolt processes have same Memory Mapping.

Ans 02

⇒ For Reading its fine coz. do data ki kya zarrorat hai just read it from one.

⇒ But For writing it's very Bad if ek process kuch change karega toh dusra mn bhi automatically changes hojaegi.

⑦ Copy ON write (COW)
we need hardware support for this

⇒ Abb hoga ye jab tk ye read kar raha hoga, its fine, lekin ye jese e write karne ki koshish karega us tym, isko ek new data allocate kardiya jaega. :)

⑧ Increment counts on current directory and open files & ? ? ?

⑨ Set child state to "ready to run".

10.

(10) Wait for the scheduler to run the process.

↓

→ scheduler ki marzi hai konse process ko pehla run karna hai

→ Hosakta hai parent pehla run hojai, maybe child. :)

Exp

```
int main()
{   int pid;
    pid = fork();
    if (pid==0)
        printf("child");
    else if (pid==-1)
        printf("Error");
    else
    {
        printf("Parent of %d", pid);
    }
}
```

you are the child

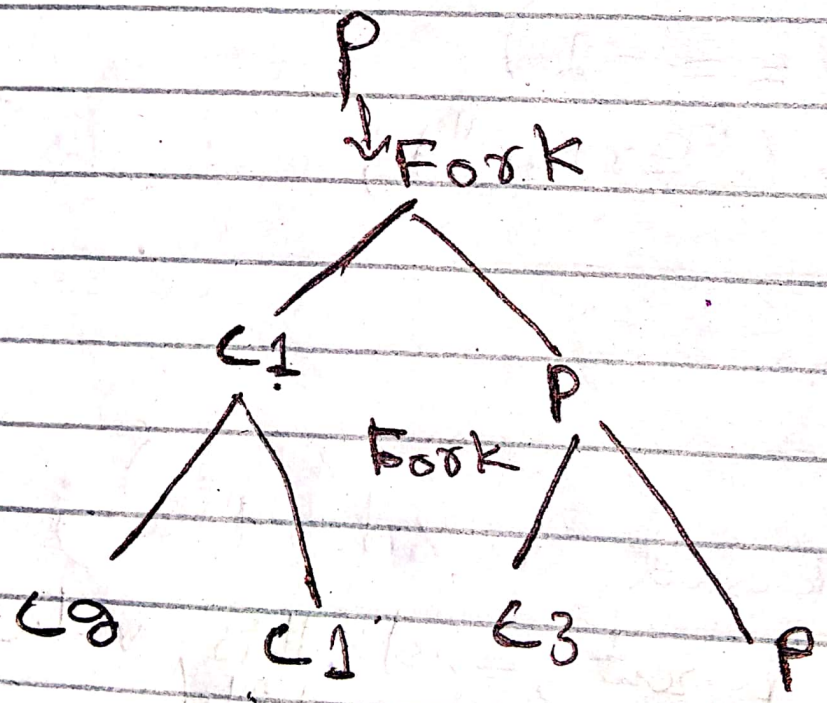
You re the parent, and this value is the pid of the child

any Non zero integer

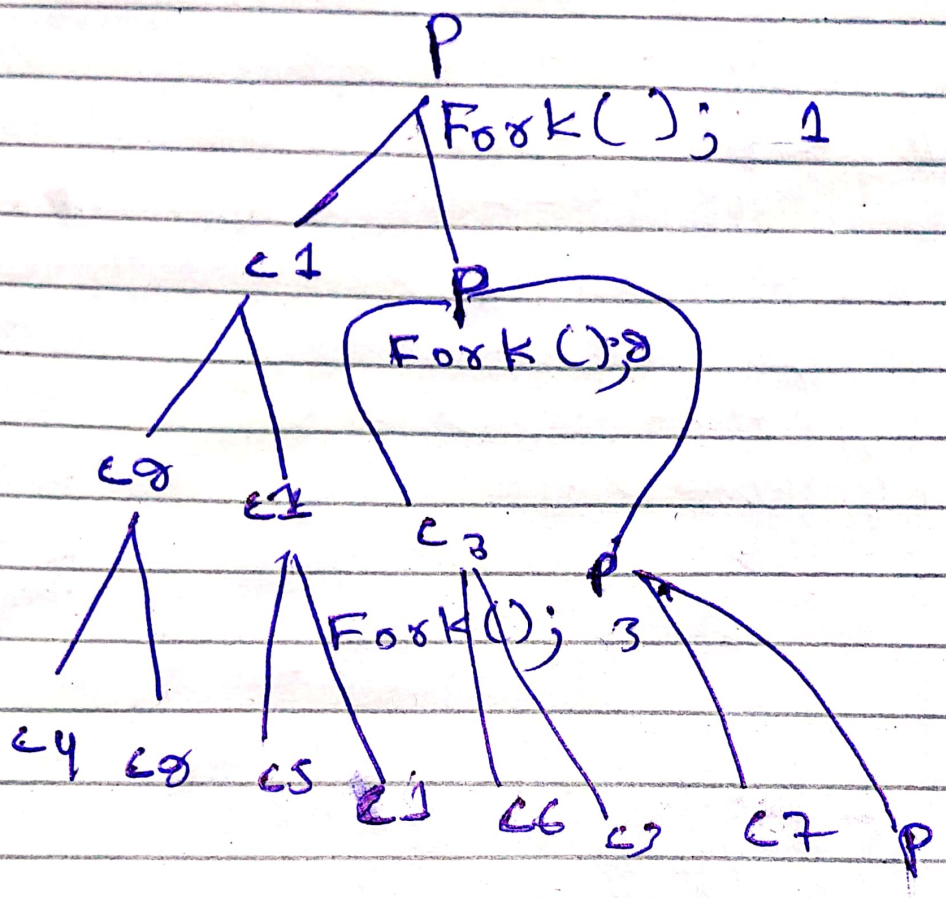
Mn parent hu, lekin apko pid child ko bhej gay hai. :)

(Get Smashers)

```
main()
{
    Fork();
    Fork();
    printF("hello");
}
```



```
Fork();
Fork();
Fork();
printf("Hello world");
```



Child process, $2^n - 1$

Hello world will be printed 2^n!