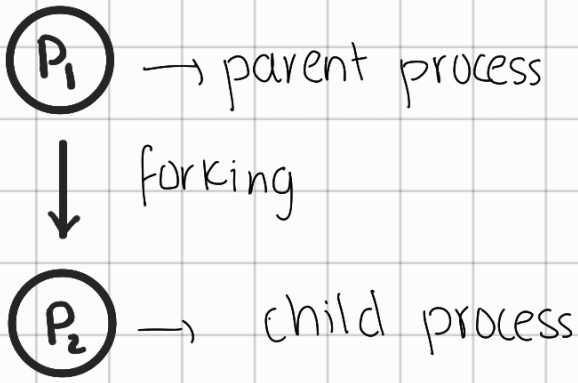
lecture 3: Process Creation

How the process is created? Unix.



the **main process** in the unix is **Init → pid=1**

- Initially when the **system boots**, a process **(init) with pid = 1 is started.**

Je n'appelle Jules

- In unix, the **process are created by cloning/ duplication** => one process duplicated

P1 becomes parent.
P2 child

- int fork();

```
#include <unistd.h>
int fork();
```

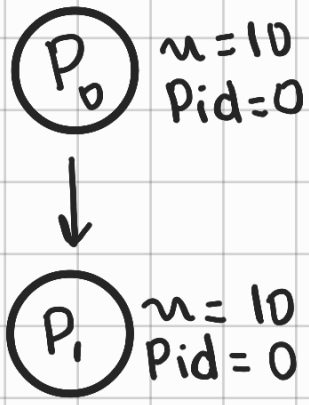

```
int fork()
```

- -1 → failure
- 0 → (child)
- <> 0 → (parent) pid of child.

Ex:

```
void main(){
  int x, pid;
  x = 10;
  if (pid = fork())
    x = x+5;
  printf("x = %d /n", x);
}
```

parent



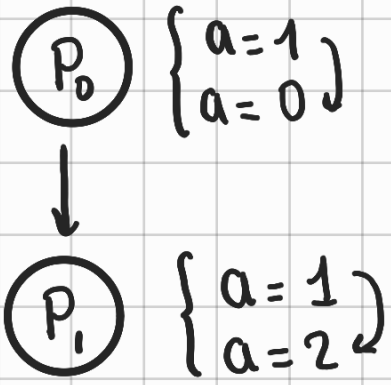

```
if (Pid = fork())
  ① pid = fork();
  ② if (pid) = if(pid!=0)
```

```
void main(){
  int a = 1;
  if(!fork()){
    a++;
    printf("%d\n", a);
  } else
    a++;
  printf("%d\n", a);
}
```

child.

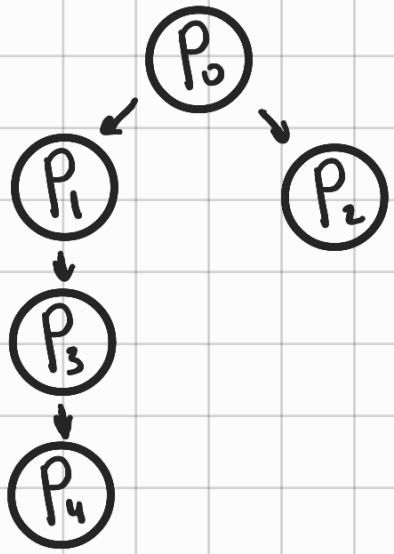
output:		
2	0	2
0	2	2
2	2	0

```
if (!fork())
  → pid = fork()
  → if (pid == 0)
```

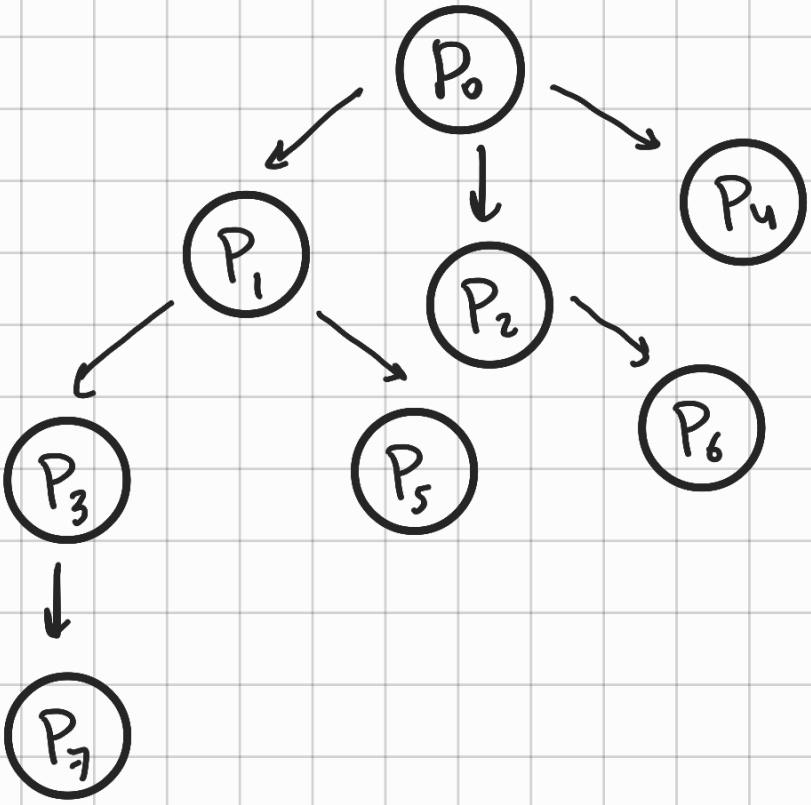


Ex:
```
void main(){
    if(fork())
        fork();
    else if (!fork())
        fork();
}
```

if (fork() => Pid = fork() => P_1
if (Pid <> 0) => P_0.



Ex:
```
void main(){
    fork();
    fork();
    fork();
}
```



Process Identification

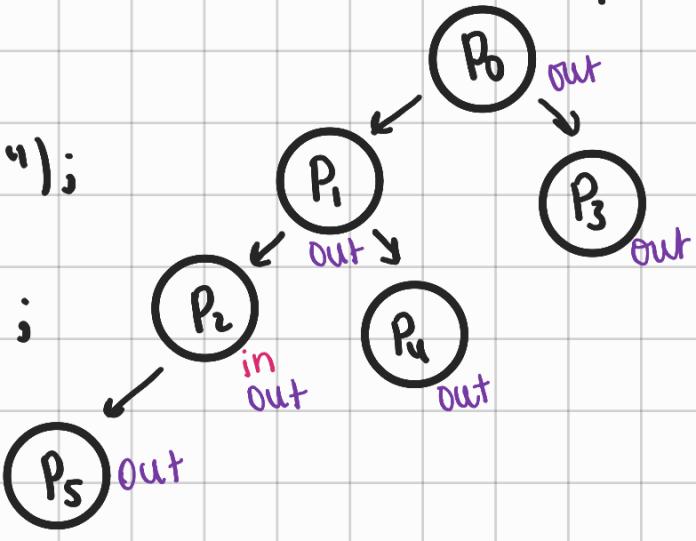
- int getpid()
  returns the pid of the calling process.

- int get ppid()
  returns the pid of the parent of the calling proces.

Extra Exercises:

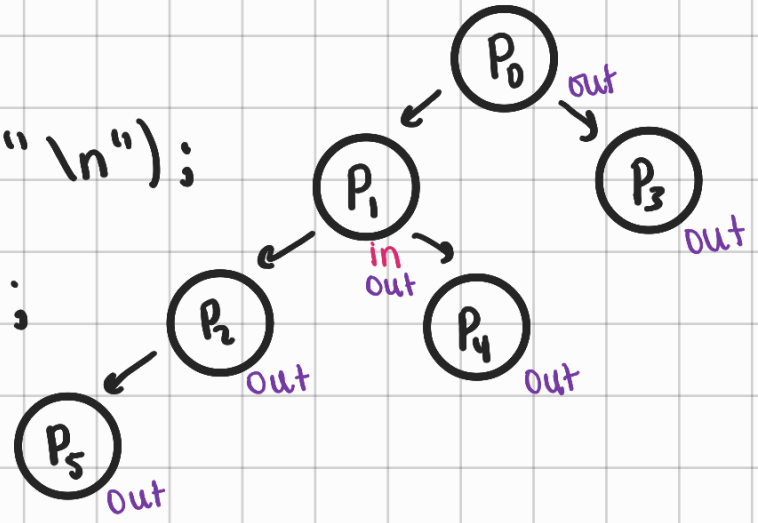
Ex:
```
void main() {
    if (!fork())
        if (!fork())
            printf("I'm in \n");
    fork();
    printf("I'm out \n");
}
```

draw tree + output



-
```
void main(){
    if (! fork())
        if (fork())
            printf ("I'm in" \n");
    fork();
    printf("I'm out \n");
}
```

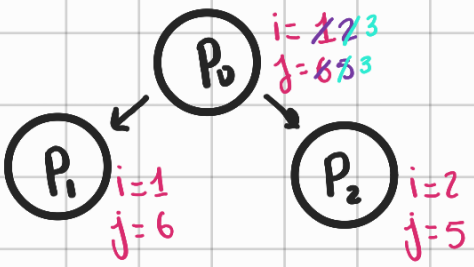
draw tree + output



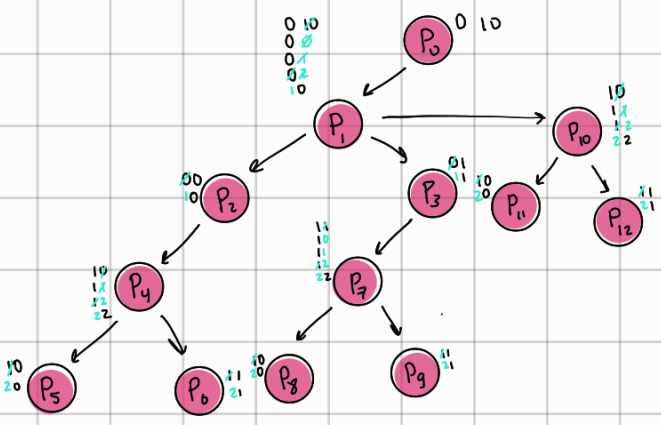
N.B: Break

-
```
void main(){
    int i,j=6;
    for(i=1; i<j; i++){
        if (!fork())
            break;
        else
            j=-i;
    }
    printf("i=%d, j=%d\n", i, j);
}
```

tree of process + output



Ex:
```
void main(){
    int i=10, j=10;
    for(i=0, i<2, i++){
        if (fork()) · if parent break
            break;
        for(j=0, j<2, j++){
            if (!fork()) if child break
                break;
        }
    }
    printf("%d, %d\n", i, j);
}
```



output: 0:0 10 4:22 8:20 12:21

1:12 5:20 9:21

2:10 6:21 10:22

3:11 7:22 11:20