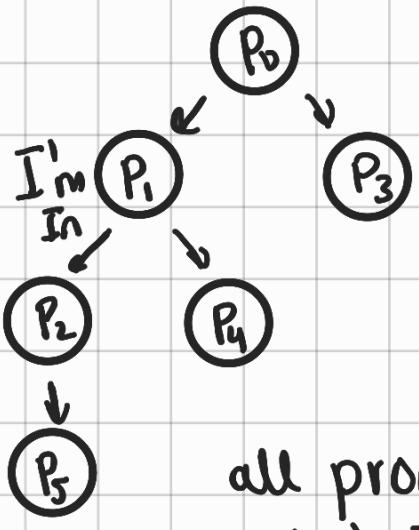
Lecture 5:

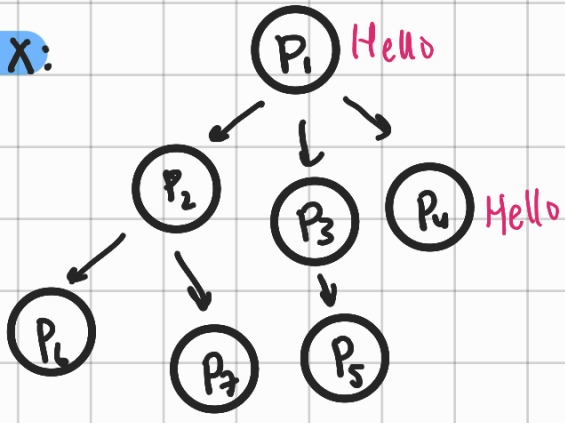
Ex:

```
void main() {
  if (!fork())
    if (fork())
      printf("I am In\n");
  fork();
  printf("I am Out\n");
```



all processes print I'm out.

Ex:



write the code that generates the following tree.
in addition, just the two processes P_1 and P_4 should print the message "Hello".

```
void main() // P1
  printf("Hello\n");
  if (fork()){ // P2
    if (fork()){ // P3
      if (!fork()){ // P4
        printf("Hello");
      }
    else{ // on P3
      fork(); // P5
    }
  else{ // on P2
    if fork() // P6
      fork(); // P7.
  }
```

Ex:

Write a program where a process creates N child processes $P_1, P_2, \ldots, P_N$. The parent process should print "Hello" when at least M process child have terminated.

The value of N and M should be given as command line argument.

```
void main (int argc , char * argv[])
  int N, M;
  N = atoi (argv[1]);
  M = atoi (argv [2]);
  int i, j;
  for (int i = 0 ; i < N ; i++) {
     if (!fork) {
        sleep (10);
        exit (0);
     }
     while( j<M) {
        pid = wait (null);
        j++;
     }
     printf ("Hello \n");
  }
```

Ex:

```
void main(){
  int x, i=2, j=7;
    while (i<j){
      x=fork();
      if (x==0 && i%2==0)
        break;
      if (!x=0)
        j--;
      i++;
    }
}
```