Lebanese University
Faculty of Sciences I

Partial 2018/2019
Duration: 1 hour

I3303 - INFO 324
Operating System II

Problem I

8 points

Given the following C program under Unix

```
void main(){
      if(fork()){   //P0
            printf("A");
            if(fork()){   //P0
                  wait(0);
                  printf("B");
            }
            else   //P2
                  printf("C");
      }
      else   //P1
            if(fork()){
                  wait(0);
                  printf("D");
            }
            else   //P3
                  printf("E");
}
```

1. Which are the possible outputs of this program?
2. Add to this program few codes such that the only possible output be EDCBA.

Problem II

8 points

Given the following program:

```
int f(int j){
      if (j == 0) return 1;
      return (fork() || fork()) && f(j-1) ;
}

void main(){
      if(f(2))
            printf("test\n");
}
```

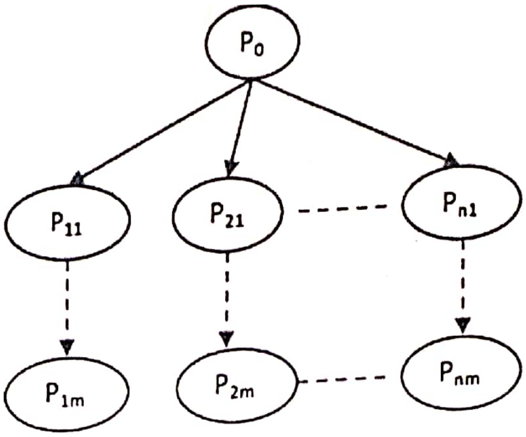
Draw the tree of processes generated by the program and show the output results for each process.

Problem III

14 points

Write a c program under Unix that creates the tree of processes in the figure under the following criteria:

- The number of levels (m) and the number of each process in each level (n) are given to the main process from the command line.
- Process P_0 waits for a time t before the creation of all the processes in the first level. After the creation of level1, process P_{11} should creates its son P_{12} before giving the turn to P_{21} that creates its son P_{22} and so on.
- The last process for each level (for example P_{n1} in level 1) should give the hand to the first process in the next level (for example P_{12}) and so on.



Scanned by CamScanner

oblem I : (8 pts)

1)

A	A	A	A	A	E	E	E	A	~~E~~ E
C	C	C	E	E	D	A	A	E	A
E	B	E	C	D	A	C	D	C	C
B	E	D	B	C	C	B	C	D	D
D	D	B	D	B	B	D	B	B	B

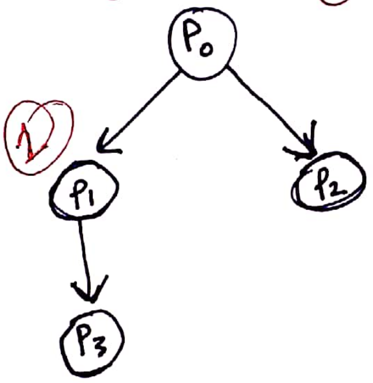
(4) = (1) (1) (1) (1)

$P_0 \rightarrow A$

B after C $\left[\begin{array}{l} P_2 \rightarrow C \\ P_0 \rightarrow B \end{array}\right.$

D after E $\left[\begin{array}{l} P_3 \rightarrow E \\ P_1 \rightarrow D \end{array}\right.$

A before B & C



(2)

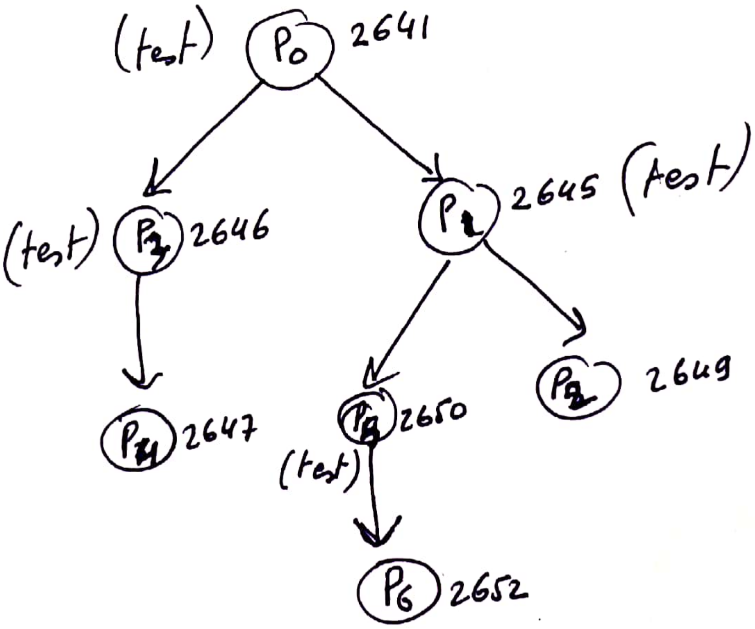
2) Impossible

Problem II (8 pts)

4→test,

4→tree

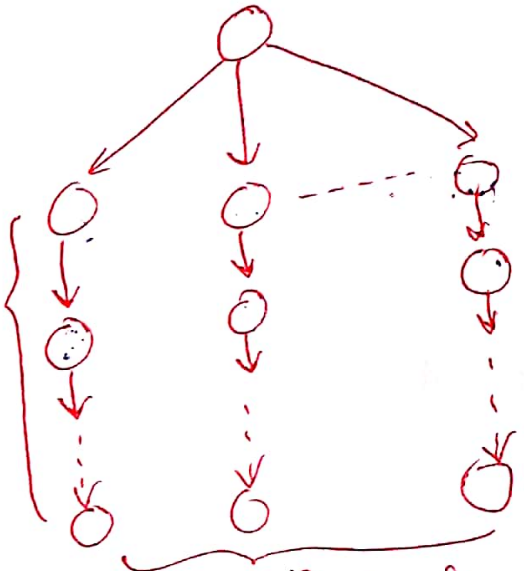
4+4



(14 / 15)

```
#include <stdio.h>
#include <unistd.h>
#include <signal.h>
#include <stdlib.h>
int fd[2],j=0,m,pid1,child;
void handle(int sig){
 printf("i received the signal %d\n",sig);          } → (1)
 printf("the values of j and m are %d %d\n",j,m);  }
 ~~signal(SIGUSR1,handle);~~
}
void main(int argc, char *argv[]){
  int n,i,pid;
  printf("my pid is %d\n",getpid());
  pipe(fd);
  m=atoi(argv[1]);  } → (1)
  n=atoi(argv[2]);  }
  signal(SIGALRM, handle); } → (1)
  signal(SIGUSR1, handle); }
  alarm(2); } → (1)
  pause();  }
  for (i=0;i<n;i++) { →
    pid=fork();            (2)
    if(pid){
      write(fd[1],&pid,sizeof(int)); →
      ~~close(fd[1])~~;
    }
    else
      break;  → (1)
  }  // end for

  if(!pid) pause(); → (1)
  if (pid && i==n){
      read(fd[0],&child,sizeof(int));  →
      close(fd[0]);                     →  (2)
      kill(child,SIGUSR1);              →
      pause();   → close(fd[1]);
  }
  printf("Hello\n");
  while(j<m-1){
    if((pid1=fork())){  → (1)
(2)  read(fd[0],&child,sizeof(int));  →
     write(fd[1],&pid1,sizeof(int));  →
     kill(child,SIGUSR1);  →
     break;
    }
    else pause(); → (1)
   j++;
  }
while(1);
} //end main
```



parent creates first level childs and put their pids into the pipe (4)

Parent awake the first child (3)

creation of each column (4)

(3) + (4) + (3) + (4)

- handle
- parameter

first level

awake 1st child

creation each column