3.5

BS SE 5A

Quiz 1

Course: Operating Systems	Course Code: CS 2006
Section: BSE-5A	Total Marks:10
Name: Jayyab Kamran Sami	RollNo: 22i-2505

Question 1: [5 Marks]

Write a C program where the parent process creates three child processes using fork(). Each child process should print its own PID and exit with a different exit code (e.g., 1, 2, 3). The parent process should wait for each child to terminate and print the exit status of each child.

(Hint: you can use WEXITSTATUS(status) macro to get exit status)

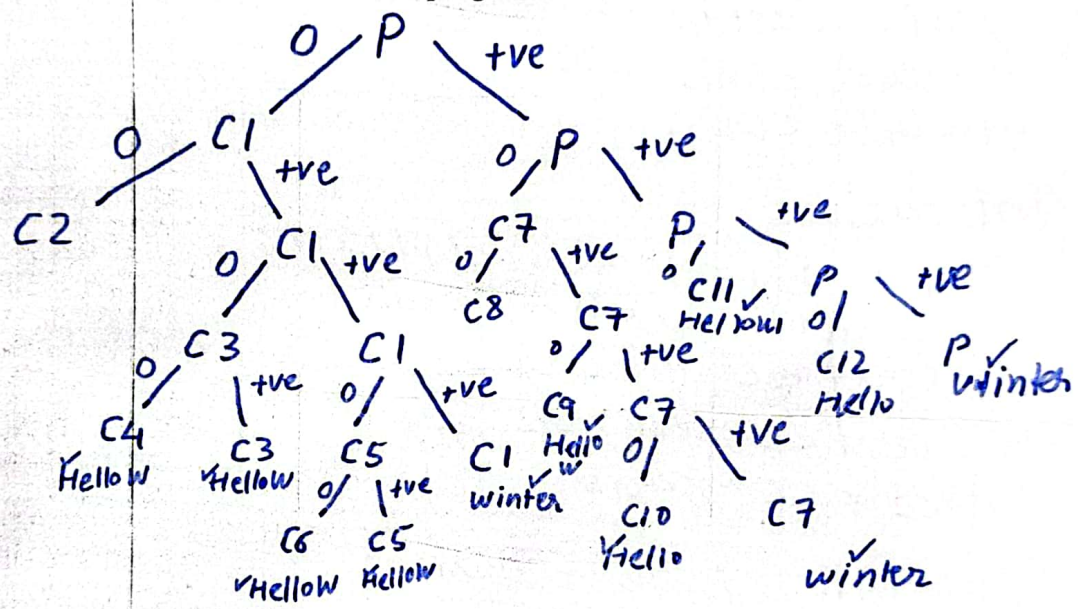
```
#include <stdio.h>
#include <unistd.h>
#include <types/sys.h>
int main() {
    int id1 = fork(); int id2 = fork(); int id3 = fork();
    wait(); wait(); wait();
    cout << "Process 1 " << id1 << endl;
    cout << "process 2 " << id2 << endl;
    cout << "Process 3 " << id3 << endl;
    int status = wexit status(& status)
```

0

Question 2: [2.5+2.5 Marks] Consider the following code:

```
if ((fork() && fork()) || fork())
{if( fork() && fork())
{ printf ("Winter is coming"); }
else
{ fork();
printf ("Hello World"); }
}
```

How many times "winter is coming" and "Hello World" will be printed on the screen? Also draw the Process Tree of the given program.



3 times "winter is coming"

8 times "Hello world"

3.5

Quiz 2

par fd[0] — fd2[0]
par fd[1] — fd2[1]

Course: Operating Systems | Course Code: CS 2006
Section: BSE-5A | Total Marks:10
Name: Tayyab Kamran Sami | RollNo: 22i-2505

Question 1: [10 Marks]

Imagine a scenario where a parent process interacts with a child process to create a guessing game. Initially, the parent process prompts the user to input an integer number. This number is then transmitted to the child process through a pipe. Subsequently, the child process will compare that number with an already stored secrete number. If the number match, then the child process communicates this information back to the parent process through the pipe. In the case of mismatch, the child process will also inform parent about failure and then the parent process prompts the user again to input a number and transmits it to the child process. This iterative process continues until the child process either successfully guesses the number or exhausts the maximum limit of 5 attempts. Given the skeleton code below you are required to complete the missing portion of the code.

```
int main {
    int fd1[2]; int fd2[2]
    int rv = pipe(fd1); int rv2 = pipe(fd2);
    int num; int cpid = fork(); if(cpid==-1){ cout<<"failed"; return }
    for(int i=0; i<5; i++){
      if(cpid>0){
      cout<<"Enter a number"<<endl;
      cin>>num;
      close(fd2[1]);
      write(fd1[1], num, 8);
      close(fd1[0]); int buff
      read(fd2[0], buff, 8);
      if( buff[0] ==1){
          cout<<"Number gussed"<<endl;
          return;
      }cout<<"Incorrect Number"<<endl;
      }
    else if(cpid==0){
        close(fd1[1]);
        close(fd2[0]);
      int secret=1122;
      int check;
     read(fd1[0], check, 8);
      int flag;
```

```
if (check == secret)
  { flag = 1;
    write (fd2[1], flag, 8)
  }
  else{
    flag = 0;
    write (fd2[1], flag, 8);
  }
}}
cout << " 5 limits exhausted "<< endl;
return 0;
}
```

10

Quiz 3

Course: Operating Systems | Course Code: CS 2006
Section: BSE-5A | Total Marks:10
Name: Tayyab Kamran | RollNo: 22i- 2505

Question 1: **[10 Marks]**

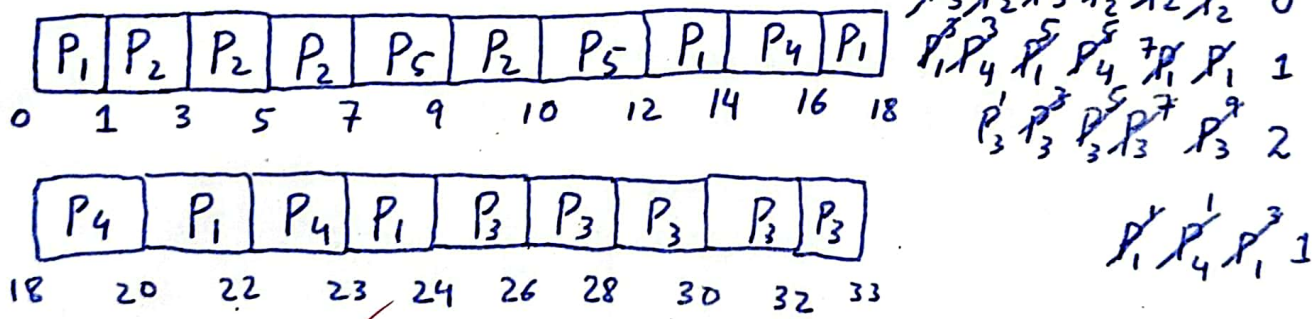
Consider the following set of processes with their respective arrival times and burst times:

Process	Arrival Time	Burst Time	Priority
P1	0	8	1
P2	1	7	0
P3	2	9	2
P4	3	5	1
P5	5	4	0

Round Robin Scheduling with Priority Preemptive (lower the number higher the priority)

Upon arrival of higher priority process lower priority process will be preempted, same priority processes will be executed through RR based on FCFS order in ready queue.

Using a time quantum of 2 units, determine the turnaround times, and waiting times for each process and there averages (with complete calculations). Show the Gantt chart as well (mandatory).



TAT $P_1 = 24$
TAT $P_2 = 9$
TAT $P_3 = 31$
TAT $P_4 = 20$
TAT $P_5 = 7$

Avg TAT = 18.2

W.T $P_1 = 16$
W.T $P_2 = 2$
W.T $P_3 = 22$
W.T $P_4 = 15$
W.T $P_5 = 3$

Avg WT = 11.6

Quiz 4

Course: Operating Systems
Section: BSE-5A
Name: Tayyab Kamran Sami

Course Code: CS 2006
Total Marks:10
RollNo: 22i-2505

Question 1: **[10 Marks]**

A parking garage can accommodate up to 50 cars. Each car entering the garage must wait if the garage is full, and cars can only enter or exit one at a time due to limited space at the entrance. Create a synchronization solution with **binary and counting semaphores** to:

- Allow cars to enter and leave safely, ensuring only one car is moving in or out at any given moment.
- Track the availability of parking spaces so that cars entering the garage wait if it's full.

You can assume that there are two functions carEnter(int car_id) and carExit(int car_id) that are used, you need to define these functions, on successful entry or exit you will print ("car 123 successfully Enter or Exit the garage")

```
full = 0; empty = N; //where N is the number of empty space
Semaphore check = 1;        int N = 50

void carEnter(int car-id)
{ do {
    wait(empty);
    wait(check);
cout<<"Car "<< car-id <<
 "successfully enter the
 garage"<<endl;
   signal(check);
   signal(full);
  }
  while(1)
}
```

```
void carExit(int car_id) {
do{wait(full);
   wait(check);
cout << "Car " << car-id <<
" successfully exit the garage";
   Signal(check);
   signal(empty);

}
while(1)

}
```