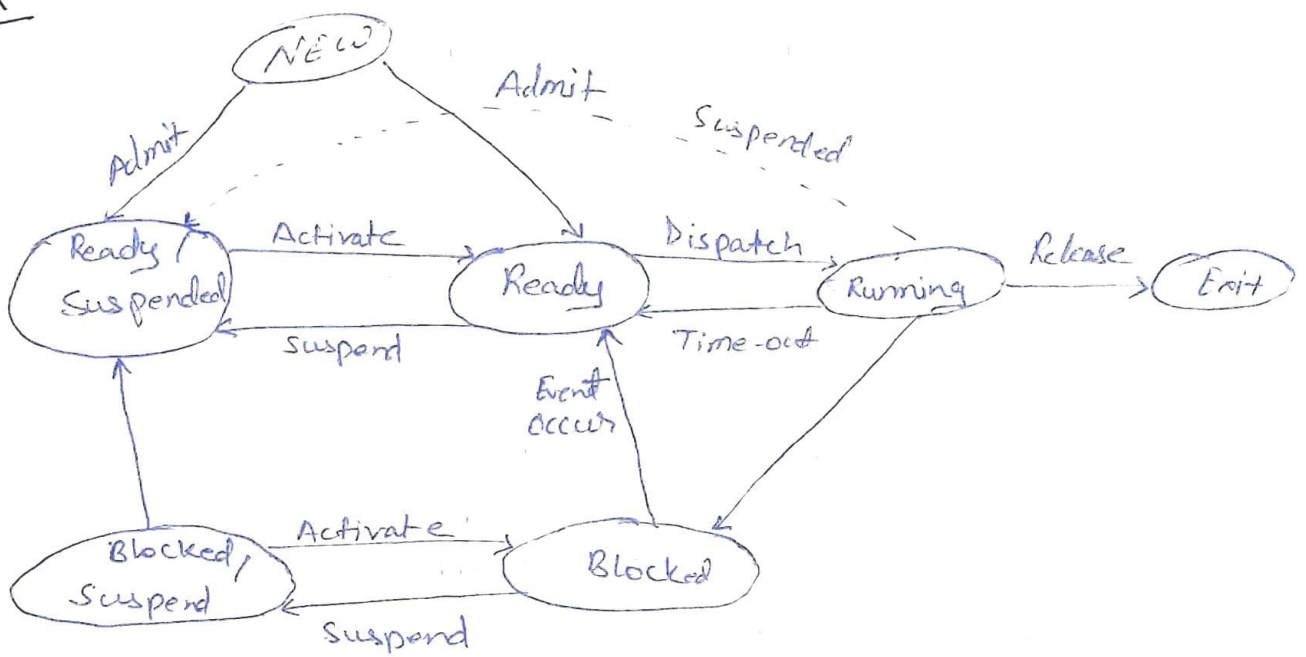
Name :- SHISHU Assigment-2 Reg:-2020CA089

OS

Ques 1

Sol



• Process State Transition Diagram with Suspend State.

Suitable Solution is :→

Ready State means less swapping also fairest that there will be no starvation. Ready suspend means that the process (in ready queue) and this is more time consuming. It does gurantee that the heigst priority process are executing.

Ques 2

Sol

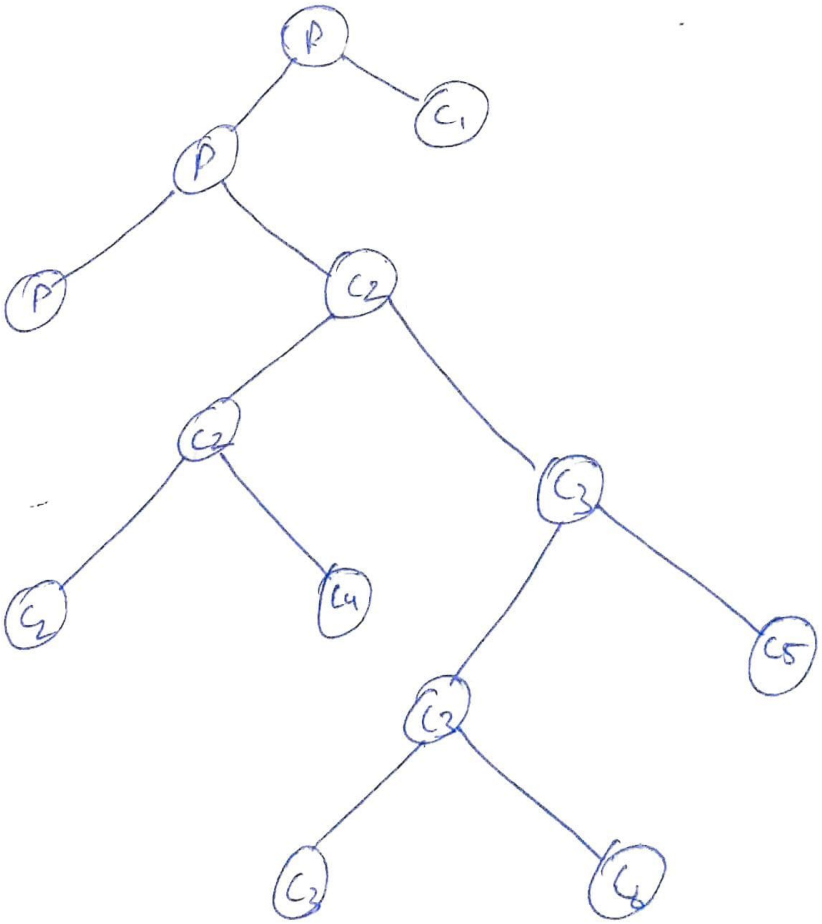
```
#include< stdio.h>
#include < unistd.h>

int main ( ) {
   if (fork() && (!fork())) {

        if (fork() || fork()) {
            fork();

        }
   }
   printf(" 2");
   return 0;
}
```

Output 2 2 2 2 2 2 2



Explation :-

1) Fork will create two process one parent P (has process id of new child) and nother one is child C_1 (process id = 0)

2) In if statement we are using AND operator and in this case if first condition is false that it will not evaluate second condition and print 2. Parent process P check for second condition and create two new process (one parent P and other is child C_2). In second condition we are using NOT operator where return true for child process C_2 and it execute inner if statement.

3) Child C_2 again Create two new process (one Parent C_2 and Child C_3 and we are using OR operator which evaluate second condition when first condition is false Parent C_2 execute if part end create two process (one parent C_2 and child C_4) where as child C_3 checked for second condition and create the new process (one parent C_3 and child C_5)

4) Parent C_3 enters in if part and further create the new process (one parent C_3 and child C_6)

Ques 3

Solu If we running a trival program (constant time complexity) in a seperate thread, The overhead of creating the thread exceeds the task performed by then, thus decreasing performance when compared to single threaded alternative.

Example

(1) Trivial operations on a list of numbers Multi threading cant speed up the operation Since the time taken by the Operation is constant and other elements of the list may or may not wait for the previous to finish.

(2) Allocating memory to a set of data variable - Allocating memory is a very fast task and the overhead of creating multiple threads to process separate blocks of variables excess the Performance gained by multithreading.

Ques 4

Solu (a) when number of Kernal threads is less than no. of processor then some of the processor would remain idle since the scheduled map only kernal thread to processors and not user-level threads to processors.

(b) when no. of kernal threads allocated exactly equal to the no. of processors then it is possible that all of the processor might be executed simultaneously However when Kernal thread block

inside the kernal the corresponding processor would remain idle.

(c) The number of kernal threads allocated to the program is greater that the number of processors but less than the number of user-level thread.-

Sol:- When the kernal threads are more than processor, a blocked kernal thread could be swapped out instead of another kernal thread that is ready to execute. This will inverse the utilization of multiprocessor system.