Important: This is an open book test. You can use any books, notes, or paper, but not exchange anything with other students. You are not allowed to use any electronic/communication devices, including a calculator. *Do not log into the computer during the test. Switch off your cell phones.* Any calculations and rough work can be done on the back side of the test pages. You will lose five points for not writing your name.
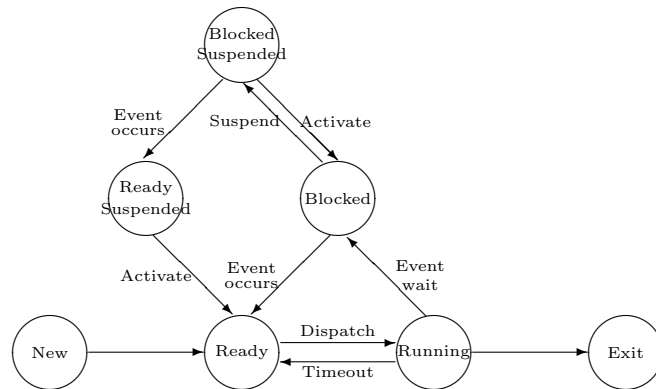
1. [6 pt] Explain the difference between an interrupt and a trap. How do you deal with multiple interrupts?

2. [6 pt] Explain the difference between monolithic kernel and microkernel.

3. [6 pt] Why does the following picture have two blocked states?

4. [6 pt] Consider a computer with N processors in a multiprocessor configuration.

   (a) How many processes can be in each of the Ready, Running, and Blocked states at same time?

   (b) What is the minimum number of processes that can be in each of the Ready, Running, and Blocked states at one time?

5. [10 pt] Consider the following program:

```
const int    n = 50;
int          tally;

void total()
{
    int count;
    for ( count = 1; cont <= n; count++ )
        tally++;
}

void main()
{
    tally = 0;
    cobegin
        total();
        total();
    coend;
    printf ( "%d\n", tally );
}
```

(a) Determine the proper lower bound and upper bound on the final value of the shared variable `tally` output by this concurrent program. Assume processes can execute at any relative speed and that a value can only be incremented after it has been loaded into a register by a separate machine instruction.

(b) Suppose that an arbitrary number of these processes are permitted to execute in parallel under the assumption of part (a). What effect will this modification have on the range of final value of `tally`?

6. [6 pt] What is a *daemon* in Unix? How can you create one?

7. [6 pt] What is the difference between a semaphore and a condition variable? Explain how the wait and signal operations are handled differently by those two.