Programming Assignment 2

Mitchel Fields

1. Thirsty Threads

```
/* a[0] for water, a[1] for ice, a[2] for cup */
semaphore a[3] = 0;
semaphore server = 1;
Server(void) {
    int i,j;
    While(TRUE){
        i = random(3); /*x returns a random integer 0, 1 or 2 for i */
        j = random(3); /* returns a random integer 0, 1 or 2 for j */
        if (i != j) { /* i and j must be different */
            Wait(server);
            k = 3 - (i+j); /*the drinker with the k-th ingredient
                                is identified*/
            Signal(a[k]);
        }
    }//end of while
}//end of Server

Drinker(int r) {
    /* r indicates which ingredients this drinker has */
    While(TRUE){
        Wait(a[i]);
        Drink( );
        Signal(server);
    }
}//end of Drinker
```

2. Lazy Dentist

```
/* The first two are semaphores are mutexes
(only 0 or 1 possible)*/
Semaphore dentistReady = 0
Semaphore seatCountWriteAccess = 1
```

```
/* if 1, the number of seats in the waiting room can be
incremented or decremented*/
/* the number of patients currently in the waiting room,
ready to be served*/
Semaphore patientReady = 0

/* total number of seats in the waiting room*/
int numberFreeWRSeats = N
```

Dentist Method:

```
def Dentist():
    while true: // Run in an infinite loop.
                /* Try to acquire a patient:
                if none is available, go to sleep*/

        wait(patientReady)

        /* Awake: try to get access to modify
        # of available seats otherwise sleep*/
        wait(seatCountWriteAccess)

        /* One waiting room chair becomes free*/
        numberFreeWRSeats += 1

        /* Doctor is ready to consult.
        signal(dentistReady)

        /* Dont need the lock on the
        chairs anymore. */
        signal(seatCountWriteAccess)

        // (Talk to patient here.)
```

Customer Method:

```
def Customer():
    /* Run in an infinite loop to
    simulate multiple patients */
    while true:

        /* Try to get access to the
```

```
waiting room chairs. */
wait(seatCountWriteAccess)

/* If there are any free seats:*/
if numberFreeWRSeats > 0:

    /* sit down in a chair */
    numberFreeWRSeats -= 1

    /* notify the dentist, whos waiting
    until there is a patient */
    signal(patientReady)

    /* dont need to lock the
    chairs anymore */
    signal(seatCountWriteAccess)

    /* wait until the dentist is ready */
    wait(dentistReady)

    /* (Consult dentist here.)*/

else:
    /* otherwise, there are no free seats; tough luck
    /* but dont forget to release the
    lock on the seats! */
    signal(seatCountWriteAccess)

    // Leave without consulting the dentist.
```