CS 376	Operating Systems	Test 1
Name:	Fall 2000	Max Pts: 43

**Important**: This is an open book test. You can use any books, notes, or paper. *Do not log into the computer during the test*. Any calculations and rough work can be done on the back side of the test pages. You will loose five points for not writing your name.

1. [6 pt] Enumerate two differences between interrupts and exceptions. Also give an example of each.

2. [8 pt] What are zombie processes in Unix? Can you kill them? What is the reason for their existence? What are the resources used by them?

3.	[5 pt] What is the difference b	between a process and a daemon in Unix?	
4.	[5 pt] In fork, join, and quites this such a requirement?	t primitives, we remarked that the join primitive is indivisible.	Why
4.	[5 pt] In fork, join, and quiis this such a requirement?	t primitives, we remarked that the join primitive is indivisible.	Why
4.	[5 pt] In fork, join, and quitis this such a requirement?	t primitives, we remarked that the join primitive is indivisible.	Why
4.	[5 pt] In fork, join, and quist this such a requirement?	t primitives, we remarked that the join primitive is indivisible.	Why
4.	[5 pt] In fork, join, and quitis this such a requirement?	t primitives, we remarked that the join primitive is indivisible.	Why
4.	[5 pt] In fork, join, and quits this such a requirement?	t primitives, we remarked that the join primitive is indivisible.	Why
4.	[5 pt] In fork, join, and quitis this such a requirement?	t primitives, we remarked that the join primitive is indivisible.	Why
4.	[5 pt] In fork, join, and quits this such a requirement?	t primitives, we remarked that the join primitive is indivisible.	Why

5. [10 pt] Does the following code satisfy critical section problem for two processes? Explain your answer.

```
extern bool flag[2] = {false, false}; // Shared with initialization as shown
extern int turn = 0;
                                       // Shared with initialization as shown
process p ( const int i )
                                       // i is 0 or 1 to identify process
    while (1)
        flag[i] = true;
        while (flag[1-i])
            if ( turn == 1-i )
            {
                flag[i] = false;
                while ( turn == 1-i );
                flag[i] = true;
            }
        critical_section ( i );
        turn = 1 - i;
        flag[i] = false;
        remainder_section ( i );
    }
}
```

6.	[5 pt] What is the difference between monolithic kernel and microkernel architecture? Why does microkernel run slower compared to monolithic kernel? What kernel architecture is used in Unix, Linux, and Windows NT?
7.	[4 pt] Give two examples (each) of applications in Unix that utilize synchronous and asynchronous messaging primitives.