**Problem Sheet for B16, Operating Systems, Wood, Trinity 2013**

1.) Recursion can be used to compute the nth Fibonacci number in the following way:

```
int fib(int n) {
   if(n==0)
      return 0;
   if(n==1)
     return 1;
   return fib(n-1) + fib(n-2);
}
```

a) What is the largest value of n that could be computed on a machine with 1MB memory if the OS pushes, including function parameters, 32 bytes to the stack for every function call?

b) What could be done to increase the largest computable value?

2.) Describe an efficient mechanism by which an OS can implement interprocess communication. Describe a distinct, less efficient mechanism too.

3.) Suppose a, b and c are three arrays of integers in a C program. Why, when N is large, might

```
for (j = 0; j < N; j++)
   for (i = 0; i < N; i++)
         c[i][j] = a[i][j] + b[i][j];
```

take a thousand times longer than

```
for (i = 0; i < N; i++)
   for (j = 0; j < N; j++)
         c[i][j] = a[i][j] + b[i][j];
```

on the same machine?

4.) Use semaphores to solve the producer comsumer problem. Assume the existence of a single reader, single writer, a character array buffer, and two semaphores; one labeled empty, the other labeled occupied. Other variables may be instantiated as needed.

5.) On the slide titled “Blocking Lock,” currently page 62, a subtle thread synchronization problem can happen. What is it?

6.) A multithreaded server processes web requests and streams audio data to 100 clients simultaneously.

a) Assuming a 100Mb/sec network interface card what is the maximum achievable audio stream rate to each client assuming a 3% packet header network overhead and perfect sharing of the network and CPU?

b) Describe using diagrams and text the data flow, scheduling, interrupt, and switching behavior of a straight threads OS running on a single core CPU required to serve each of these clients.

c) Note in particular the OS services required and used. Also, compute the order of magnitude of the minimum CPU clock speed required to saturate the network if memory reads and writes take two CPU clock ticks, device I/O is PIO-style, and saving context including register state takes five CPU clock ticks.

7) Assume 8-Kb pages, how big is a page table for a 64-bit architecture? What are some strategies for dealing with this problem?