Spencer Shimko
CMSC 421
HW 1

PART A: (50 points)

1. (5x3 = 15 points)

Log into the linuxserver linux1@gl.umbc.edu.

a) What operating system does it run? What is the kernel version it runs?
OS = Linux
Kernel = 2.4.26 w/ SMP

```
bash-2.05a$ uname -a
Linux linux1.gl.umbc.edu 2.4.26 #1 SMP Thu Apr 22 13:38:42 EDT 2004 i686
unknown
```

b) Is this a Batched system?
Yes this is a batched system. It has the memory layout of a batched system where the OS is seperated from user programs.

c) Is this a Multiprogrammed system?
Yes this is a multiprogrammed system. It has many processes (programs) running (or waiting to run [see time-sharing below]).

d) Is this a Time-sharing system?
Yes this is a time-sharing system. Processor/resource time is divided between many processes.

e) Is this a Multiprocessor system?
I believe this is NOT a multiprocessor linux system with some caveats (read below). The output below shows that it is a HyperThreaded P4/Xeon class processor. This is why there are 2 CPU's listed. But hyperthreading is NOT true multiprocessor, there is still only 1 physical processor and 1 core. The other alternative is that Hyperthreading is disabled. If this is the case then it IS a multiprocessor machine.

Output:

```
[sshimko1@linux1 ~]$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 3.06GHz
stepping        : 9
cpu MHz         : 3052.254
cache size      : 512 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
```

```
fpu             : yes
fpu_exception   : yes
cpuid level     : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe cid
bogomips        : 6094.84

processor       : 1
vendor_id       : GenuineIntel
cpu family      : 15
model           : 2
model name      : Intel(R) Xeon(TM) CPU 3.06GHz
stepping        : 9
cpu MHz         : 3052.254
cache size      : 512 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 2
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe cid
bogomips        : 6094.84
```

2. What are system calls? What is their purpose? How are they handled? (15 points)

The method user processes use to request that the kernel perform privleged operations. It is handled like a software interrupt (typically 0x80).

3. Problems 2.5 on pages 52-53 in the text book. (10 points)

a) privileged
b) unprivileged
c) privileged
d) privileged
e) privileged

4. Problem 4.6 on page 127 from your text book. (10 points)

This algorithm can be altered to allow storage of ***n*** items in buffer by adding a variable that tracks the current size of the buffer. This variable “cursize” will be initialized to 0. The producer will increment this variable whenever it adds to the buffer. The consumer will decrement this variable whenever it removes an item from the buffer. The key is in the producer's while loop. An different condition is used in the consumers while loop. It runs:

```
 while ( cursize == BUFFER_SIZE )
      ;
```

This prevents the producer from adding to a full buffer and allows full use of the ***n*** sized buffer.

PART B: (50 points)

Familiarize yourself with the /proc filesystem in the Linux OS (eg at a minimum look at the man pages for proc). Log on to linux.gl.umbc.edu and explore the /proc directory. This directory provides an easy way to get a lot of information about the current system.

The file /proc/meminfo contains information about the free and used memory in the system. The unix command "free" uses this file to generate its output. Write your own version of the command "free" in C. Run the command "free" at your promt to see the sample of what the output should look like.

Submitted seperately (see README and Makefile)