COP 4600 – Midterm Examination - 1

Date: October 2, 2013

**Name: ………………………………………………………………………………………………………….**

Instructions:

* This exam is open book and open notes. Allotted time is 75 minutes.
* Note that the points add up to 100 + 20 bonus points.

# Problem 1 (10 points)

**Circle the correct answer:**

A \_\_\_\_ can be used to prevent a user program from never returning control to the operating system.

A) portal

B) program counter

C) firewall

D) timer

Embedded computers typically run on a \_\_\_\_ operating system.

A) real-time

B) Windows 7

C) network

D) clustered

When a child process is created, which of the following is a possibility?

A) The child process runs concurrently with the parent.

B) The child process has a new program loaded into it.

C) The child is a duplicate of the parent.

D) All of the above

**Solutions:**

**D**

**A**

**D**

**(each of them worth 3 pts)**

# Problem 2 (10 points)

Explain why an operating system can be viewed as a resource allocator (3 sentences)

**Various resources are necessary to run a program: memory, processor time, hard disk space, files, network bandwidth.**

**-Must mention at least 2 for full grade**

**-Must mention scarcity or resources or competition for resources.**

# Problem 3 (10 points)

Give two arguments why a printer driver should run in kernel mode, and two arguments why it should run in user mode (2+2 sentences)

**+ higher performance**

**+ easier to program**

**- safety issues: do you trust the manufacturers?**

**- reliability: failure should not bring down the OS**

# Problem 4. (10 points)

Describe why direct memory access (DMA) is considered an efficient mechanism for performing I/O. (3 sentences)

**The processor can do other stuff while the I/O transfer is happening.**

# Problem 5 (10 points)

Consider that you are designing Windows 9. Give two arguments in favor of a microkernel organization, and two arguments against it (2+2 sentences)

**+ more modular, smaller number of components to run in kernel mode**

**+ micro-kernel components easier to write (they are smaller)**

**- more difficult to program the interaction between kernel modules**

**- a monolithic kernel is more efficient, less kernel / user switches**

# Problem 6 (10 points)

Give two arguments for the use of virtual machines, and two arguments against it. (2+2 sentences)

**+ Ability to run multiple operating systems on the same hardware**

**+ Better separation between users (compared to different accounts)**

**+ More efficient use of the hardware**

**- Performance problems**

**- Complexity**

**- Wasted resources**

**- Hardware is cheap**

# Problem 7 (10 points)

Give three reasons why it is advantageous to run a web server as a multi-threaded process (3 sentences)

**+ A new request can be serviced without waiting for the previous one to finish**

**+ Compared to multi-process: a lot of the code is shared**

**+ More efficient use of the server hardware**

# Problem 8 (10 points)

Discuss the way the in which the Round Robin algorithms behavior is impacted by the size of the time quantum? (about 4 sentences)

**-Too short: too much switching overhead. Higher responsiveness.**

**-Too long: virtually no overhead. Not interactive, resembles batch processing.**

# Problem 9 (20 points)

Consider the following code segment. Explain what will happen and how many processes you will have (a drawing would help):

**for(int i = 0; i!=3; i++) {**

**fork();**

**}**

**Let us call the first process A. The first process will create 3 processes A-1, A-2, A-3 (all of which are the same, but start with i=1, 2, 3 respectively. A-1 will create two processes: A-1-2, A-1=3. A-2 will create a process A-2-3. On its turn A-1-2 will create process A-1-2-3.**

**So we will end up having 8 processes: A, A-1, A-2, A-3, A-1-2, A-1-3, A-2-3, A-1-2-3.**

# Problem 10 (20 points)

You are a software developer. For each of the following things say whether

x. it happens or not, you as a software developer cannot do anything about it

y. it is bad thing, and you as a software developer can do things to avoid it

z. it is good thing, and you as a software developer can do things to make it happen

For each of the following,

* Specify if it is x,y or z and explain in one sentence why it is so.

(a) atomic operation

**X – having some operations atomic is useful for parallel programming, but you cannot ensure it in software**

**(1 pt for answering z)**

(b) critical section

**X – it comes from the meaning of the program. You can try to reduce its size in the program, but you cannot completely avoid it.**

(c) deadlock

**Y – you can avoid deadlock through various techniques**

(d) mutual exclusion

**Z. – you can introduce it to handle critical sections**

(e) race condition

**Y. – Different parts of the program will always run faster or slower. You can, however, avoid situations where the program will have different outcomes function of that.**

(f) starvation

**Y – you can avoid deadlock through various techniques, such as aging**

(g) Discuss what would change above if you are a hardware developer.

**You gain the ability to implement atomic operations. Some of the problems cannot be solved in hardware (deadlock, starvation).**