COSC 4330 FIRST MIDTERM FEBRUARY 21, 2005

This exam is **closed book**. You can have **one** page of notes. UH expels cheaters.

1. Answer in **one sentence** to each of the following questions: (6×5 points)

(a) Why do most operating systems on the market continue to use ***monolithic kernels***?

Because monolithic kernels are faster than microkernels since they do not incur any extra context switch overhead.

(b) What is the purpose of the **`signal(...)`** system call?

To "catch" a signal sent by another process, that is, to specify which action should be taken when it receives that signal.

(c) Why is ***memory protection*** never implemented in software?

Because each and every memory reference must be checked.

(d) What is the value returned by the **`fork(..)`** system call?

It returns zero in the child process and the process ID of the child in the parent process.

(e) Which event(s) will move a process from the ***waiting state*** to the ***ready state***?

A process goes from the waiting state to the ready state once its system request has completed.

(f) What happens when a process does a **`wait(...)`** on a child process that has ***already terminated***?

It immediately returns the process to the ready state.

2. ***Advantages and disadvantages****: you will get no credit if you mention a disadvantage when an advantage is asked and vice versa.* (6×5 points)

(a) What is the main advantage of ***modular kernels****?*

They are more extensible than regular monolithic kernels without being slower than them.

(b) What is the main advantage of ***multithreaded servers*** over servers that do not use threads?

They can process several requests in parallel without incurring the overhead of forking new copies of themselves.

(c) What is the main advantage of the ***symmetric organization*** for multiprocessor operating systems?

Since the kernel can run of any processor, it will never become a bottleneck.

(d) What is the main advantage of ***dual-mode CPUs***?

We can prevent user processes from performing I/O operations without kernel intervention, thus allowing the kernel to control accesses to user data on disk.

(e) What was the main advantage of ***time-sharing systems*** over ***batch systems***?

They increase the productivity of programmers (and allow new types of applications such as word processing, electronic mail and so on).

(f) What is the main disadvantage of ***user-level threads***?

Whenever one thread in an address space does a blocking system call, the kernel will move the whole address space to the waiting state thus blocking all threads.

3. How many lines will the following program print out? (5 points)

```
main() {
      fork();
      printf("Hello!\n");
      fork();
      printf("Goodbye!\n");
} // main
```

Answer: _SIX_ lines.

4. How will the following code fragment affect **stdin, stdout and stderr**? (3×5 points)

```
int fda, fdb;
fda = open("alpha", O_RDWR | O_CREAT, 0640);
fdb = open("beta", O_RDWR | O_CREAT, 0640);
close (2);
dup(fda);
close(0);
dup(fdb);
```

stdin is redirected to file "beta" _____

stdout is unchanged _____

stderr is redirected to file "alpha" _____

5. A newly created process waits for the CPU for 50 ms, gets the CPU for 40 ms, does a system call and gets again the CPU for 60 ms. List the state visited by the process starting with the ***New*** state. (5×2 points)

(a) New _____

(b) Ready _____

(c) Running _____

(d) Waiting _____

(e) Ready _____

(f) Running _____

6. Would UNIX have had the same impact if its kernel had been written in assembly code? **(**2×5 points)

No because it would not have been possible to port it to other architectures and it would have been much harder to modify the kernel.