CMSC 412 Midterm #2 (Spring 2014)

1.) (20 points) Define and explain the following terms:

a) Super page

A larger page (4MB vs. 4KB on x86-32) to allow better use of TLB slots (since one TLB entry can reach 4MB vs. 4KB).

b) Extent based file system

A filesystem using groups of consecutive blocks together as an extent. Used to reduce seek time in accessing blocks of a single file. Often benefits from pre-allocation (user indicating file size at creation time) to know how to size the extent.

c) Bankers Algorithm

A way to avoid deadlock. Simulate request and verify the existence of a safe sequence before allowing the request to complete.

d) Inverted page table

Page table using a hash function to store virtual->physical mapping information. The size if proportional to the physical size of memory rather than the virtual address range.

2.) (20 points) Memory Systems

a) (7 points) Consider an x86-64 like architecture with a hardware TLB (with an access time of 1 ns for a hit or miss), a four level page structure (4 levels of 512 entries each). Memory access time is 100ns. Assume all memory fits into RAM (i.e. no page faults to disk). What TLB hit rate is required to achieve an effective access time (EAT) of 105ns?

Miss = 501ns; hit = 101ns, so 101ns always . thus 400x=4ns for miss is 1% thus a 99% hit rate is required.

b) (6 points) What is a typical use for the accessed/referenced bit on the x86 memory management unit (MMU)? What sets the bit? What clears it?

Used to implement a clock like approximation of LRU page replacement. Set by the processor (haware mmu) on use, and cleared by the OS (kernel) when the clock sweeps.

c) (7 points) Why is a base/limit register pair less flexible than a conventional page table?

It requires a contiguous allocation of memory.

3.) (20 Points) Synchronization: You need to synchronize the process of getting and making coffee in the CS department lounge. In the department, there are two types of coffee urns (regular and decaf) and one coffee maker. If someone goes to get a cup of coffee of one type, and that pot is empty they make a new pot of that type of coffee. Provide a solution using semaphores (include variable declarations and initial semaphore values) to the coffee problem that ensures:

- Only one person at a time is taking coffee out of the decaf urn
- Only one person at a time is taking coffee out of the regular urn

- Only one urn is on the coffee maker at a time
- If one type of coffee is being made, people requesting coffee of the other type can get it if it is available.

You may assume there exists a function emptyUrn(bool checkDecaf) that returns true if the coffee urn indicated by the parameter is empty.

Semaphore decaf = 1, regular = 1, maker = 1;

While …

```
If (wantsDecaf) {
    P(decaf);
    If (emptyUrn(wantsDecaf)) {
        P(maker);
        // make coffee
        V(maker);
    }
    // pour decaf
    V(decaf);
} else {
    P(regular);
    If (emptyUrn(wantsDecaf)) {
        P(maker);
        // make regular coffee
        V(maker);
    }
    // pour regular
    V(regular);
```

4.) (20 points) File Systems

a) (6 points) Why does the default installation of NTFS not check directory permissions, but only checks file permissions?

There is a time overhead on file open to lookup and check the ACLs on each directory

b) (8 points) What are the differences between hard links and symbolic links?

Hard links are two files that share an inode, symbolic links store a linked file name as its contents and have a field indicating the file is a symbolic link.

c) (6 points) Consider a file system with 4KB blocks and 64 bit block indexes, Each inode has 10 direct blocks, 1 indirect block, and 1 double indirect block. What is the largest file possible that can be stored?

512 entries per block

10x4096 + 1 x 512 x 4096 + 1x 512 x 512 x 4096 = $10x2^{12} + 2^{21} + 2^{30}$ bytes – about 1 GB

5.) (20 points) Project

a) (6 points) Explain why it is possible to use a single LDT for all user processes in the project once the paging code for project #4 is done.

LDT defines the base and limit registers for segmentation. In P4 all user processes are in the range 2GB to 4GB so can all share the same LDT entry.

b) (6 points) Even though it is mapped in the range above 2GB (user space), the APIC and IO-APIC pages are not accessible from user mode. Why?

Protection is required so user processes can't change the interrupt handling via the APIC and IO-APIC.

c) (8 points) If we wanted to move the APIC and IOAPIC pages to an address range outside user space (2GB to 4GB), explain the changes needed to GeekOS to do this.

Just map it to a different location (1 and 2 pages before 2GB in the kernel space would work well. Then need to change used of the APIC and IO-APIC to know about the new mapping location (in GeekOS this is just changing the global variables APIC_Addr and IO_APIC_Addr.