Com S 352 **Homework 1** **John Collin Vincent**

1.12)

a)

1)

users may be able to view and take other users files and information.

2)

if permissions for users/groups are not configured properly users may be able to edit system configuration files and compromise the whole environment

b)

you can have a high level of security in a time shared environment with proper permission settings, but the fact that you are regularly letting third parties perform operations on the time shared machine creates a inherently more vulnerable environment than a dedicated machine for each user.

1.14)

users who are collaborating on a project, or have a need for a uniform environment that can be changed by another team to consistently mirror the production environment will find a time shared machine more efficient than multiple dedicated machines.

1.15)

symmetric multiprocessing is where there are multiple CPUs that all have the same architecture and are managed by a single OS with some shared memory. asymmetric multiprocessing is where there are multiple CPUs that may or may not have the same architecture and may or may not be running the same OS, they all have a method of communication between them but do not have a shared memory pool.

(advantages)

1) users can run multiple processes at a time
2) users can run different architectures together like a cpu and gpu
3) speed of execution

(disadvantages)

its hard to synchronize the processors and make them work together in a consistent and stable way.

1.19)

an interrupt is used to signal to the cpu that a hardware process has finished and is ready to be handled. An interrupt is different from a trap in that a trap is thrown by a user program and is often the result of a division by 0 or an invalid memory access. yes traps can be thrown intentionally and are often used to switch context mode.

1.20)

a)

The cpu sends a request to the DMA controlled to initiate the transfer.

b)

An interrupt is triggered to alert the cpu when the memory process is complete.

c)

this process only affects operations from user programs that are dependant on that memory that is being transferred or operations that would use the DMA controller.

2.13)

1) parameters are passed in a register to the OS
2) parameters are stored in a block and the address of the block is passed in a register
3) parameters are pushed onto the system stack

2.15)

1) managing the file systems used on the different drives
2) managing permissions to different files
3) managing the hardware resources so read and writes happen in a stable order
4) allocate memory for files
5) managing what files are currently being accessed by what user.

2.16)

the benefit is that it's easier for user programmers to interact with devices since they use the same system calls as the files and the calls in this interface are well defined and understood. the difficulty comes from trying to shoehorn lots of different devices into these calls that were designed for files.

2.18)

the first is message passing, this is much safer as the programs are more separated and they have much less influence over each other. the second option is shared memory, this is much faster but also more dangerous because one process can change memory that another process was still dependant on if the synchronization is not done properly leading to somewhat unpredictable behavior.