Javier Escareno

Professor Li

CSCI 114

February 5, 2023

4a.)   The Operating System should allocate time on the processor based on task priority. The higher level of importance the task is, the more processing time should be given to it. If their a multiple tasks ready to go at the same time it should still in my opinion go by order of importance regardless of how long either task may take.

4b.)   Operating systems should allocate physical memory to applications based on when the application asks to use the memory. It would run on a first come first serve type of system. If a set of applications do not fit at the same time in the memory then applications currently not being used should have their memory swapped to the local storage disk so the in use application can run.

4c.)   An operating system should allocate its disk space by first come first serve storage the data files that are stored right next to one another until there is no longer any room. The outcome of the policy would likely lead to either mass file deletion everytime storage is full or the buying of more storage every so often.

9.)   I would design a system to before the update is scheduled to do a mandatory manual storage backup of all data saved and once that is done then could the update take place so that no data is lost if the machine crashes during the update.

1.)   For x86 sepereate stacks are used for kernel and user processes to keep the system sequre. It does this incase during the interruption a privilege or stack switch is inadvertently turned on

during the interrupts that happens while doing the user process and will save the kernel stack in place without saving the previous current state before the interrupt.

7a.)   The iret function would be used in the operating system when the operating system comes across an issue that requires immediate attention.

7b.)   When an app executes return from an interruption its cpu performs a check for privileges then restores the code segments, registers, stacks, and pointers upon others.