CSE 536 – Advanced Operating Systems

Name: Mohith Sadashiva Reddy **ASU ID: 1233347260**

Question 1:

a) Purpose of init and sh processes in xv6:

i. **Init:**

- The init process is the first user-space process that runs after the kernel has finished its initialization. It is responsible for starting the user environment and acts as the parent process for all other user processes.

ii. **Shell:**

- The sh process is the shell, which is an interactive command-line interpreter. It allows users to execute commands, run programs, and manage the file system. The shell reads user input (commands), interprets them, and executes them. Commands could be file operations, launching programs, or built-in shell commands. The shell allows users to launch processes, redirect input/output, and manage job control

b) Role of uvmalloc and loadseg during process creation:

i. Uvmalloc()

- Uvmalloc function is used to allocate virtual memory during process creation. It is used to create new pages in the process's page table, mapping them to physical memory. It also grows the process's virtual address space by adjusting the sz variable to reflect the new total size of the process's memory.
- It passes parameters such as:

  pagetable → process' page table which maintains mappings from virtual addresses to physical addresses

  sz → size of the process' memory

  ph.vaddr + ph.memsz → new size of the memory after allocating enough space to hold the section defined by the program header

  flags → this is to set the read, write, execute permissions for the newly allocated memory

ii. Loadseg()

- This function loads the contents of the program section from the binary file into the process' memory and copies the relevant part of the executable into the virtual memory space. It uses the file descriptor ip to read the section's contents from disk starting at the offset ph.off and copies them into the process's memory starting at the virtual address ph.vaddr. The section size loaded into memory is determined by ph.filesz.

Question 2:

a) The mkfs.c file is responsible for creating the initial file system for xv6. It writes out the inode table, the file data, and the bitmap representing free/used blocks. The PSA is usually allocated by reserving a section of the disk during the filesystem creation process.

The page swap area (PSA) is reserved within the filesystem image in the mkfs.c file by doing the following:

- The swap area is defined using the nswap variable, which is set to the value of PSASIZE. This indicates the number of disk blocks reserved for the swap area.
- The calculation of the total number of metadata blocks is adjusted to include the blocks reserved for the swap area.
- The swap area is allocated a specific range of blocks on the disk, starting after the log blocks and before the inode blocks. This is achieved by adding the superblock structure
- The swap area starts at block swapstart and extends for nswap blocks. This area is reserved for swapping out pages and is not used for file storage.