Operating Systems Design - 19CS2106A

Home Assignment - CO-I

Mode of submission: Post Hand written scanned documents in LMS

Submission date: on or before 18th October, 2020

1. Illustrate log design. why logging metadata updates ensure the recovery of a file system after a file system crash. What are the fields of the super block and give its use?

2. The file system can be viewed as graph with i-nodes as nodes and directory entries as links. Using link can a program create cycles in the UNIX v6 i-node graph? If so, show a sequence of commands that creates a cycle. If not, how does v6 prevent cycles?

3. (i). Directories provide a hierarchical structure. Explain how Names are mapped to inodes recursively.
   (ii). Consider a file system similar to the one used by UNIX with indexed allocation. How many disk I/O operations might be required to read the contents of a small local file at /a/b/c? Assume that none of the disk blocks are currently in main memory.

4. Write a Program in which the reader and writer processes carry out two-way communication using two pipes. This program should opens two pipes, pipe1 and pipe2, and creates a child process. The child process sends a message to the parent process using pipe1 and reads the parent’s response from pipe2. The parent does it the other way round. Whatever data the parent and child processes read from their respective pipes, they throw to standard output. show the setup in a pictorial form.

5. How does xv6 create a file? List xv6 kernel functions/algorithms and files directly or indirectly used for the execution of $ echo > a

6. Write down the algorithm for disk block allocation.

7. (i). What is the file table and per-process file descriptor table? Explain how to use them when accessing a file?
   (ii) In xv6, Explain the working of open("a/b/c", O_RDWR |O_CREATE). List xv6 kernel code functions/algorithms used.

8. Illustrate the algorithm for the Conversion of Byte Offset to Block Number in File System.
   A UNIX file system has 4-KB blocks and 4-byte disk addresses. What is the maximum file size if i-nodes contain 10 direct entries, and one single, double, and triple indirect entry each?

9. (i). List the tasks performed by the read () system call.
   (ii). Imagine a process that performs the system call: read (fd, buf, 100)
   which means that 100 bytes should be read from the file indicated by fd into the memory buffer buf. Illustrate access to disk blocks uses the buffer cache. Is it possible to read the desired block directly into the user’s buffer, and save the overhead of copying?

10. (i). Why are users never allowed to directly write to a directory? Why are file attributes not stored in the directory entry itself? why must directories be empty for before they are allowed to be removed?
    (ii). In xv6, Explain dirent structure and the working of dirloopup, dirlink functions.

Operating Systems Design - 19CS2106A

Home Assignment - CO-II

Mode of submission: Post Hand written scanned documents in LMS

Submission date: on or before 18th October, 2020

1. Illustrate UNIX system boot and processes' creation

2. In xv6, explain swtch function that does the job of switching between two contexts, and old one and a new one. The swtch function is called at two places what are they.

3. Write an algorithm for a system call that makes Parent waits and collect the termination status of a child Process.

4. Write a program that creates two child processes. Each created (child) process has to execute a separate executable program. Make the parent process wait until both child processes are completed, before being terminated.

5. In Xv6, how system call works? List the data structures, functions, and files that are to be manipulated. How do system calls relate to the OS and to the concept of dual-mode (kernel mode and user mode) operation?

6. Explain the algorithm for process termination. How the kernel treats zombies and orphans.

7. Why per-process kernel stack? Explain Scheduler () in xv6, Give the summary of entries in xv6 PCB

8. Where is usually the round robin scheduling algorithm used? What is the priority scheduling? The round-robin (RR) scheduling algorithm is designed for time sharing systems. In RR, pre-emption is added to FCFS to switch between processes. Consider the following set of processes with CPU burst time of each of them is given in milliseconds. Process CPU burst time (ms) (time required by a process) P1 15 P2 5 P3 7 P4 10 Draw the Gantt chart for RR scheduling where time quantum q = 4 milliseconds. Calculate the average waiting time and turnaround time.

9. The traditional UNIX scheduler enforces an inverse relationship between priority numbers and priorities: The higher the number, the lower the priority. The scheduler recalculates process priorities once per second using the following function:
   Priority = (Recent CPU usage / 2) + Base
   where base = 60 and recent CPU usage refers to a value indicating how often a process has used the CPU since priorities were last recalculated. Assume that recent CPU usage for process P1 is 40, process P2 is 18, and process P3 is 10. What will be the new priorities for these three processes when priorities are recalculated? Based on this information, does the traditional UNIX scheduler raise or lower the relative priority of a CPU-bound process?

10. Explain the function sched used by a process that wishes to relinquish the CPU. How does xv6 store process state? How does an OS keep time?