OS - Theory Assignment-1

T.Srivatsan – CS20BTECH11062

1. Consider 2 processes pro0 and pro1 running on CPU0 and CPU1. Let us also assume that both pro0 and pro1 use a specific data from the main memory. Both processes then store a copy of that data in their respective caches.

Let the processes perform different operations on the data in cache. Let the value of the output of the two processes be v1 and v2 such that v1 not equal to v2. The outputs values are stored in the data variable in their respective caches. Thus, the data in main memory now has 2 different values in both the caches. The last processes among the two to write the data back into the memory decides the value of that data in main memory.

2. The system calls used by printf and scanf are write and read respectively. Printf uses write which writes data from a user buffer into any data stream like a file at the lowest level of a system call. It takes 3 arguments namely the destination file id, the starting pointer of the data to be written into the file and the number of bytes to write.

ssize_t write(int fd, **const** void *buf, size_t nbytes);

It returns the number of bytes successfully written onto the file or -1 if any error occurred

Scanf uses read which is used to read data from a data stream like a file at the lowest level of a system call. It takes 3 arguments namely the locator to the file to read from, the buffer where the read data must be stored and the number of bytes to read.

ssize_t read(int fd, **const** void *buf, size_t count);

It returns the number of bytes read from the file and -1 if any error occurred

Sources:

https://courses.engr.illinois.edu/cs241/sp2012/lectures/05-syscalls.pdf

https://en.wikipedia.org/wiki/Write_(system_call)

https://en.wikipedia.org/wiki/Read_(system_call)

3. PCB's are used by Operating systems to track a process's execution status. These come in handy when the operating system wants to switch to another process before completion of the current process. The PCB's provide the register and stack data and the program counter for the process so that the OS can resume the process from the point it was paused.

The following are the main components of a typical linux PCB are as follows

Process Scheduling state: It contains the status of the current process namely if its "ready", "waiting" or "running". It also stores the priority of the current process and its total elapsed time.

Process Structuring information: It contains the process Id's of the children of the current process or the Id's of processes related to the current one in some functional way.

Pointer: Stack pointer used to restore the current position of the process when the process is switched from one state to another.

Process Number: It stores the unique process id of the current process used to identify it

Program Counter: It stores the address of the next instruction to be executed for the current process

Register: It stores the data that was present in the EPC and the general-purpose registers which stores the values of the program variables at that instant. When the PCB is loaded the current registers are loaded with the register values from the PCB so that the process can be resumed from its stopped point.

Open files list: It contains the list of the files that are kept open for the current process.

Accounting information: It contains the info regarding the amount of CPU used for a particular process and the time limits before which the OS switches process etc.

Process privileges: It specifies the system resources that the current process has access for and the resources that it does not.

Inter-process Communication information: It contains the messages and other info shared to communicate between 2 independent processes.

Sources:

https://eng.libretexts.org/Courses/Delta_College/Operating_System%3A_The_Basics/03%3A__Processes_Concepts/3.4%3A_Process_Control