Assignment 5 (Xv6-Project)

Deadline - Nov 13th

Q1:

Here we try to create a basic system-call. The first step is to extend the current proc structure and add new fields ctime, etime and rtime for creation time, end-time and total time respectively of a process. When a new process gets created the kernel code should update the process creation time. The run-time should get updated after every clock tick for the process.

To extract this information from the kernel add a new system call which extends wait. The new call will be int waitx(int *wtime, int *rtime). The two arguments are pointers to integers to which waitx will assign the total number of clock ticks during which process was waiting and total number of clock ticks when the process was running. The return values for waitx should be same as that of wait system-call.

Create a test program which utilises the waitx system-call by creating a 'time' like command for the same. When the program is executed it would be calling the waitx system call which would be giving you the wait time and the running time of the current process being executed.

General Comments :

1) The xv6 OS base code can be downloaded from https://github.com/mit-pdos/xv6-public
2) Whenever you add new files do not forget to add them to the Makefile so that they get included in the build.
3) Syscall reference, http://www.cse.iitd.ernet.in/~sbansal/os/previous_years/2011/xv6_html/syscall_8c.html#68e387cd28d033ae0ae49e18ae587b95
4) We will be taking a tutorial on this to give you a basic idea on xv6 and how to work with cscope. This would be quite helpful while navigating through the xv6 code to find things you need.
5) Make sure to have fun and gain some experience and insights on how an actual operating system works while doing the assignment ;)