CS3523 OPERATING SYSTEMS-2:

REPORT:

IMPLEMENTATION:

- In the part-1 we add a new system call setPriority to existing system calls & which was already done in previous assignments .
- We have added priority value to each system call by just adding it in the struct proc.
- Implementation of setPriority system call is done in sysproc.c . In this we take integer as argument & check whether the input is valid or not i.e. a valid priority number should be between 0 & 9 .
- If yes then we call another function change_priority which was defined in proc.c, This function takes the process id & priority value as arguments and then it assigns that priority value to that process.
- change_priority function iterates through all the processes in table & checks for a process id which matches with the input ptocess id myproc()->pid & then it updates priority of that process to priority value which is given as input.
- As by default all processes have priority value 5 . so we declare p->priority=5 for all processes.
- In the part-2 we just schedule the processes based on their priority values .
- We modify the scheduler function in proc.c file such that every time we find the process which is in runnable state & having minimum priority value i.e. high priority.

OBSERVATIONS:

- By invoking setPriority system call . I had observed that the priority of process is changed from default value .
- After the implementation of that scheduling based on priorities I had observed that the processes with low priority value i.e. high priority will be running first. & the processes with high priority value i.e. low priority will be running next. So,high priority processes are running first & then the low prority processes.
- Which indicates that the processes are being scheduled according to their priorities .

PROBLEMS:

PART-1:

- By implementing such systemcall , process can make request to OS to change it’s priority & OS will give the priority to any process who invoked setPriority system call .
- So, The problem here is any process which was not that useful right now can also gain high priority just by invoking setPriority system call & also the priority number can be choosen by the process itself.
- There will be possible starvation that some processes will have low priority & they will be suspended for ever.

PART-2:

- Now by implementing such priority based policy . OS assign priorities to the processes that invoke the system call setPriority
- This scheduling will execute the process in the order of their preferences (process having less priority number will have high priority in order of execution).
- This leads to possible starvation , As remaining processes of high priority number will be suspended all the time .
- Also by this implementation we may give more priority to some user processes than the important kernel processes.
- We solve the above issues by just assigning high priority to the process only if it is necessary .

LEARNING FROM ASSIGNMENT:

- I had learned how can a process can change it’s priority by accessing the OS through a system call .
- I had also learned how to implement the scheduling of processes based on their priorities .

Submitted by

Kodavanti Rama Sravanth

CS20BTECH11027