CS 503 HW2

Shuheng Guo

Question:1

（1）The first error is that mask = disable(); should be above the if condition of isbadpid. In the if condition curly braces, it cannot restore mask before disable the interrupt. In this case, if the current process is interrupted before make any changes to goal process’ priority, and any other process makes change to goal process’ priority, there will be a competition and ambiguity.

（2）The second error is that the code doesn’t check the value of newprio. Newprio should be no less than 1. Because all processes’ priorities are greater than 0.(Null process has priority equal to 0). f newprio is less than 1， the process with this pid will never get a chance to execute after changing the process priority.

(3) The third error is that if the process of which we have changed the priority is on ready list, we have to suspend it and ready it again. Because the ready list is a priority queue, we have to insert the process to its correct position. Otherwise, other process with less/higher priority may be executed before/after this process.

Question 2：

/\* chprio.c - chprio \*/

#include <xinu.h>

/\*------------------------------------------------------------------------

\* chprio - Change the scheduling priority of a process and return the

\* priority the process had before the change

\*------------------------------------------------------------------------

\*/

pri16 chprio(

pid32 pid, /\* ID of process to change \*/

pri16 newprio /\* New priority \*/

)

{

intmask mask; /\* Saved interrupt mask \*/

struct procent \*prptr; /\* Ptr to process' table entry \*/

pri16 oldprio; /\* Priority to return \*/

mask = disable();

/\* If pid is invalid, return SYSERR \*/

if (isbadpid(pid) || newprio < 1) {

restore(mask);

return (pri16) SYSERR;

}

/\* Obtain exclusive access to the process table and reset the \*/

/\* process's priority \*/

prptr = &proctab[pid];

oldprio = prptr->prprio;

prptr->prprio = newprio;

if (prptr->prstate == PR\_READY) {

suspend(pid);

ready(pid);

}

restore(mask);

/\* Return the priority the process had before the change \*/

return oldprio;

}