alsoNice(int n) System Call & Usage

How it works:

"alsoNice" is intercepted by the EXEC command in the shell where clock tick value is extracted. "n" and "clock tick" are interchanged.

clock tick value is passed to a system call created called "alsoNice(int clocktick)" which works similar to a fork but takes in the clock tick value. The clock tick value is passed to allocproc() which is modified from void to take clocktick. When the process is being created a new attribute is added to struct proc called clocktick which keeps track of whether or not the process needs to run for longer.

This proc p→clocktick is checked for in trap.c's trap() function which then reroutes the yield process to the same process instead of finding a new one.

Examples of calling from xv6: alsoNice *clock tick values* cmd

```
$S20 alsoNice 100 ls
$S20 alsoNice 100 alsoNiceTest
$S20 alsoNice 20 whatIf sample.txt
```

Expected results:

All the methods have been modified to display the process creation time and the first yield time. You should see the results of the process that began running and the clock tick value. lp is a method I created which prints a list of all the processes that have been created with their first yield time. While more stats have not been printed, exit time and run time can also be inferred.

```
$S20 alsoNice 20 lp
Name      pid      state    creationtime    yieldtime
 init     1        SLEEPING       0         0
sh        2        SLEEPING       5         0
sh        11       SLEEPING       30452     0
lp        12       RUNNING        30453     0

  ------------------------------------------------

  pname: lp      pid: 12        ctime: 30453    ytime: 30473

  ------------------------------------------------


  ------------------------------------------------

  pname: sh      pid: 11        ctime: 30452    ytime: 0

  ------------------------------------------------
```

Files updated

Changed de*tailed in schedulingAdditions.txt in doc/PA2*

Makefile

defs.h

lp.c

proc.c

proc.h

sh.c

syscall.c

syscall.h

sysproc.c

alsoNiceTest.c

trap.c

user.h

usys.S