Phase 1

Create a new variable pstat that is accessible by all kernel functions

Include

Moved pstat.h from include to kernel

Kernel

pstat.h <- create this file to hold the pstat definition

defs.h <- put globals here, pstat struct

proc.h <-add import at the top for pstat.h

proc.h<- add the pstat into the proc struct, so each process contains more info

Install two syscalls and write the user level program files for each

Include

syscall.h <- define 2 new entries

User level

user.h <- define 2 new entries

usys.S <- define 2 new entries

makefile.mk <- define 2 new entries

settickets.c <- create this

getpinfo.c <- create this

Kernel

sysfunc.h <- define 2 new entries

syscall.c <- define 2 new entries in the table

sysfile.c <- write sys\_settickets sys\_getpinfo

proc.c <- allproc function is where all new processes get routed through, this is where we give the process tickets, put a print statement for number of tickets

Test creating a user process

run a user program and see if the print statement in proc.c works

Phase 2

Edit the round robin scheduler

1 put a print statement before running a process

Print the number of tickets that process has

2 put a print max statement before running a process

Print the processes with the highest number of tickets

3 edit the RR scheduler to do a lottery scheduler

Select next process to run by choosing the max tickets

Phase 3

Use the test scripts