LOTTERY SCHEDULING ALGORITHM STEPS

STEP 1: ADD lotterytest in Makefile (ADD _lotterytest\)

STEP 2: in proc.c inside allocproc()
p->tickets = 10; (set default value of ticket as 10)

STEP 3: Add lottery_Total() function inside proc.c

```
int lottery_Total(void){
struct proc *p;
int ticket_aggregate=0;
//loop over process table and increment total tickets if a runnable process is found
for(p = ptable.proc; p < &ptable.proc[NPROC]; p++)
{
if(p->state==RUNNABLE){
ticket_aggregate+=p->tickets;
}
}
Return ticket_aggregate;
}
```

STEP 4: inside scheduler function declare variables

```
Int count = 0;
Int golden_ticket = 0;
Int tital_no_tickets = 0;
```

STEP 5: Now inside scheduler reset the variables to make scheduler start from the beginning of the process queue

```
golden_ticket = 0;
count = 0;
total_no_tickets = 0;
```

NOW calculate Total number of tickets for runnable processes
total_no_tickets = lottery_Total();
pick a random ticket from total available tickets
golden_ticket = random_at_most(total_no_tickets);

STEP 6: INSIDE PROC.C
find the process which holds the lottery winning ticket

```
if ((count + p->tickets) < golden_ticket){
count += p->tickets;
Continue;
}
```

STEP 7: Inside scheduler add break;

STEP 8: IN procdump.c
Add cprintf("%d %s %sc%d", p->pid, state, p->name, p->tickets);
Remove cprintf("%d %s %s", p->pid, state, p->name);

STEP 9: In proc.h
Declare int tickets;

STEP 10: IN syscall.c
Declare function as extern int sys_settickets(void);
ADD [SYS_settickets] sys_settickets,

STEP 11: IN syscall.h
#define SYS_settickets 23

STEP 12: In sysproc.c add sys_stettickets()
```
Int sys_settickets(void){
int ticket_number;
if (argint(0, &ticket_number) < 0)
{
proc->tickets = 10;
}
else{
proc->tickets = ticket_number;
}
Return 0;
}
```

STEP 13: in user.h add system call
int settickets(int);

STEP 14: in usys.S
SYSCALL(settickets)

STEP 15: Add random_at_most(total no of tickets)
```
int xn = 12;
//random no
int random_at_most(int m)
{
   int a = 56;
   int b = 43;
   xn = (a*xn+b)%(m+1);
```

```
    return xn;
}
```

STEP 16: Create lotterytest.c

```
include "types.h"
#include "stat.h"
#include "user.h"
#include "fcntl.h"

int main(int argc, char *argv[])
{
    int ticket, pid;
    if(argc < 3 ){
    printf(2, "Usage: nice pid ticket\n" );
    exit();
    }
    pid = atoi ( argv[1] );
    ticket = atoi ( argv[2] );
    if ( ticket < 0 || ticket > 100 ) {
            printf(2, "Invalid ticket (0-100)!\n" );
            exit();
    }
    settickets( pid, ticket );
    exit();
}
```