Task 1:

Inside struct proc in proc.h added:
int sched_count;
int run_ticks;

Task 2:

Inside allocproc() in proc.c added:
p->sched_count = 0;
p->run_ticks = 0;

Task 3:

Inside scheduler():
p->sched_count++;

Task 4:

Inside case T_IRQ0 + IRQ_TIMER in trap.c:
if(myproc() && myproc() -> state == RUNNING) {
        myproc() -> run_ticks++;
}

Task 5:

In syscall.h:
#define SYS_getstats 25

In syscall.c:
extern int sys_getstats(void);
[SYS_getstats] sys_getstats,

Task 6:

In sysproc.c:

```
int
sys_getstats(void)
{
  int *user_stats_ptr;

  if(argptr(0, (void*)&user_stats_ptr, 2*sizeof(int)) < 0)
  return -1;
  struct proc *p = myproc();
  int kernel_stats[2];
  kernel_stats[0] = p->sched_count;
```

```
    kernel_stats[1] = p->run_ticks;

    if(copyout(p->pgdir, (uint)user_stats_ptr, (char*)kernel_stats, sizeof(kernel_stats)) < 0)
    return -1;
    return 0;
}
```

Task 7:

In user.h:
```
struct procstats {
    int sched_count;
    int run_ticks;
};
int getstats(int *stats_array);
```

Task 8:

In usys.S:
```
SYSCALL(getstats)
```

Task 9:

Added statstest.c:
```
#include "types.h"
#include "stat.h"
#include "user.h"

int main(void) {
    int stats[2];
    int i;
    for(i = 0; i < 2; i++) {
        if(getstats(stats) == 0) {
            printf(1, "Scheduled %d times, ran for %d ticks\n", stats[0], stats[1]);
        }
        else {
            printf(2, "getstats failed\n");
        }
        sleep(10);
    }
    exit();
}
```
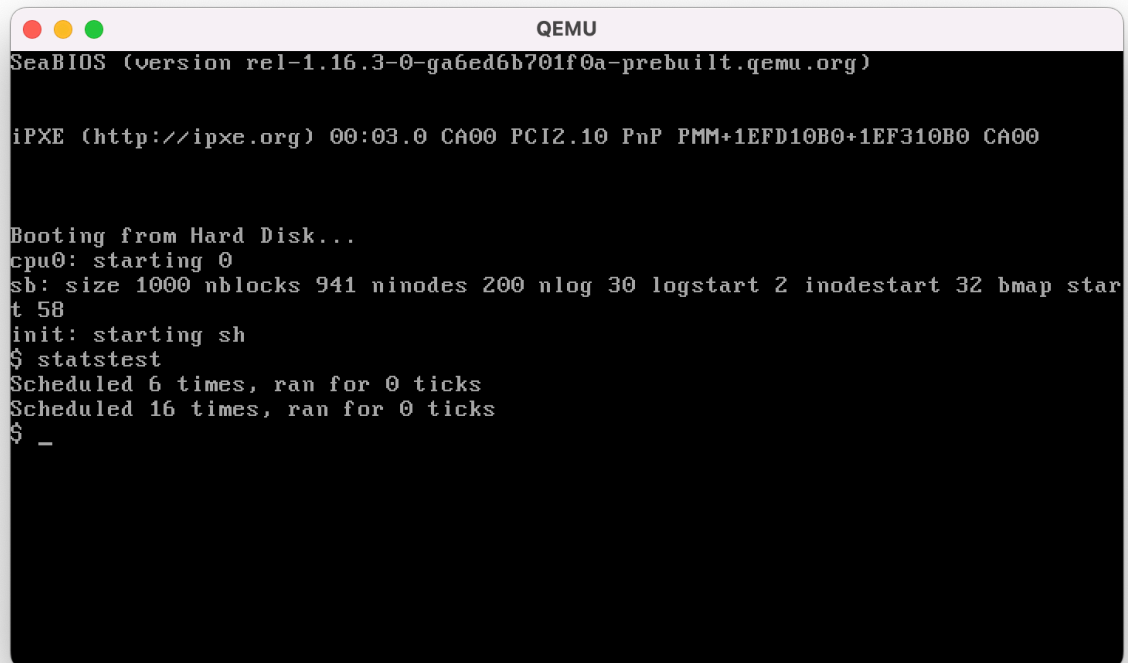
Task 10:

In Makefile:
```
_statstest\
```

```
SeaBIOS (version rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org)


iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1EFD10B0+1EF310B0 CA00



Booting from Hard Disk...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap star
t 58
init: starting sh
$ statstest
Scheduled 6 times, ran for 0 ticks
Scheduled 16 times, ran for 0 ticks
$ _
```

Screenshot of the QEMU terminal