XV6

- **Task 1 – Update proc.h — Add Statistics Fields**

  File Name: `proc.h`
  Code Added:

```
int sched_count;        // count how many time this process
                               has  been sheduled
 int run_ticks;       // count how many timer ticks this process
                       has been running
```

- **Task 2 : Initialize Counters in proc.c**

  File Name: `proc.c`
  Code Added:

  inside the allocproc () function

```
p->sched_count=0;
 return p;
```

- **Task 3 : Update Scheduler in proc.c**

  File Name: `proc.c`
  Code Added:

  inside the scheduler () function

```
p->sched_count++;
```

- **Task 4 : Update Timer Interrupt in trap.c**

  File Name: `trap.c`
  Code Added:

```
if(myproc() && myproc()->state==RUNNING){
     myproc()->run_ticks++;
   }
```

- **Task 5: Define and Map the Syscall**

  File Name: `syscall.h`
  Code Added:

```
# define SYS_getstats 25
```

  File Name: `syscall.c`
  Code Added:

```
extern int sys_getstats ( void ) ;
```

  And inside `syscalls[ ]` table:

```
[ SYS_getstats ] sys_getstats ,
```

- **Task 6 – Implement Syscall Handler in sysproc.c**

  File Name: `sysproc.c`
  Code Added:

```
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];

    // Fill kernel stats from process structure
    kernel_stats[0] = p->sched_count;
    kernel_stats[1] = p->run_ticks;

    // Copy to user space
    if(copyout(p->pgdir, (uint)user_stats_ptr,
               (char*)kernel_stats, sizeof(kernel_stats)) < 0)
        return -1;

    return 0; // success
}
```

- **Task 7 & 8: Create User-Space Interface**

File Name: `user.h`
Code Added:

```
struct procstats {
  int sched_count;
  int run_ticks;
};
int getstats ( int * stats_array ) ;
```

File Name: `usys.S`
Code Added:

```
SYSCALL ( getstats )
```

- **Task 9: Create Test Program statstest.c**

  File Name: `stststest.c`
  Code Added:

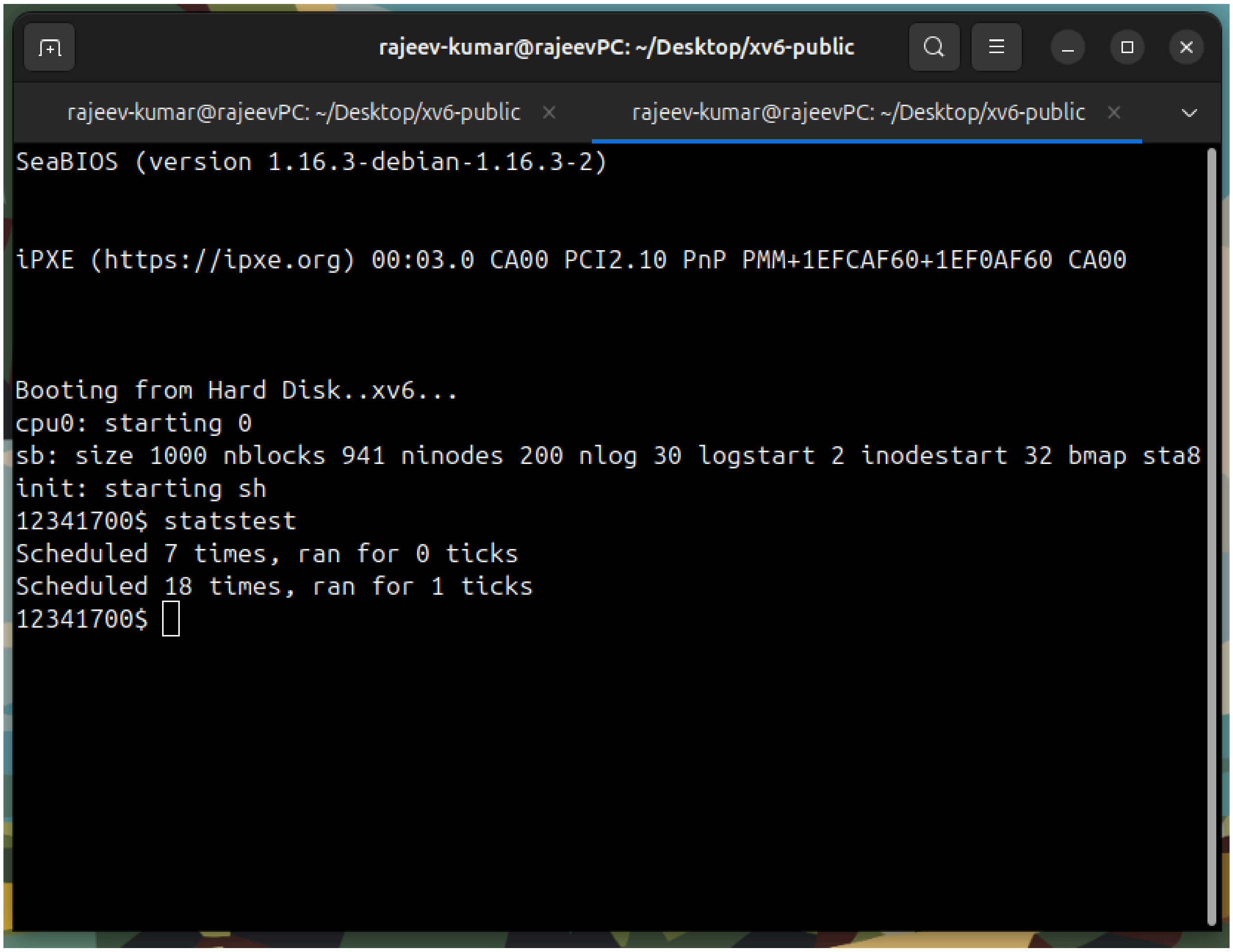
  ```
  #include "types.h"
  #include "stat.h"
  #include "user.h"

  int main(void) {
      int stats[2];   // array to hold [count, ticks]
      int i;

      for(i = 0; i < 2; i++) {
          // Call the system call
          if(getstats(stats) == 0) {
              // If success, print statistics
              printf(1, "Scheduled %d times, ran for %d ticks\n",
                     stats[0], stats[1]);
          } else {
              // If syscall failed
              printf(2, "getstats failed\n");
          }

          // Sleep for a while before calling again
          sleep(10);
      }

      exit();
  }
  ```

- **Task 10: Build and Run**

  File Name: `Makefile`
  Code Added in UPROGS:

  ```
  _statstest\
  ```

- Output :



rajeev-kumar@rajeevPC: ~/Desktop/xv6-public
rajeev-kumar@rajeevPC: ~/Desktop/xv6-public
rajeev-kumar@rajeevPC: ~/Desktop/xv6-public

```
SeaBIOS (version 1.16.3-debian-1.16.3-2)


iPXE (https://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+1EFCAF60+1EF0AF60 CA00



Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap sta8
init: starting sh
12341700$ statstest
Scheduled 7 times, ran for 0 ticks
Scheduled 18 times, ran for 1 ticks
12341700$
```