Describtion of the countTraps()

Step 1: User calls countTraps()

This will invoke a function defined in ***usys.S***

```
SYSCALL(countTraps)
```

The following macro will execute, ***eax*** means the value of SYS_countTraps should be put in the ***eax*** register.

```
#define SYSCALL(name) \
  .globl name; \
  name: \
    movl $SYS_ ## name, %eax; \
    int $T_SYSCALL; \
    ret
```

This value of SYS_counTraps comes from the definition in ***syscall.h***. Since 22 is the value of SYS_Traps, the ***eax*** register will have 22.

```
#define SYS_countTraps  22
```

The value of ***int*** becomes 64 as the value of T_SYSCALL is 64 in ***traps.h***:

```
#define T_SYSCALL       64      // system call
```

Step 2: trap function call

Define the structure **trap_kernel_statis_s** for statistics and the function **trap_get_statis** that returns the statistics

```
void trap(struct trapframe *tf) {
    trap_statis_inc(tf->trapno);
    ...
}

void trap_statis_inc(uint trapno)
{
    ...
}

int trap_get_statis(void *data)
{
    ...
}
```

Step 3: syscall function

The syscall function is defined in the ***syscall.c***, the function checks the value of the ***eax*** register. If the value is a valid trap number.

```
void syscall(void) {
 ...
}
```

The sys_countTraps function is defined in the ***syscall.c*** file.

```
static int (*syscalls[])(void) = {
    [SYS_countTraps]   sys_countTraps,
    ...
};
```

```
int sys_countTraps(void)
{
    ...
}
```