实验一报告

实现的功能

为了实现课程要求，考虑到需要在每个任务被调用时增加 syscall 计数与更新被调用时间，做了如下改动：

- 在 TaskControlBlock 中新增了 task_syscall_times 与 task_time，并在初始化 TASK_MANAGER 时初始化这两个字段。
- 当第一个任务被执行与切换到新任务时，更新当前任务时间与初始化下一个任务时间。
- 在 TaskManager 中新增相关方法，并暴露到对全局变量操作。
- 在 syscall 方法调用时增加当前任务的 syscall_times。
- 完善 process.rs 中的方法，完成课程要求。

问答题

1. 正确进入 U 态后，程序的特征还应有：使用 S 态特权指令，访问 S 态寄存器后会报错。 请同学们可以自行测试这些内容（运行 三个 bad 测例 （ch2b_bad_*.rs） ）， 描述程序出错行为，同时注意注明你使用的 sbi 及其版本。

```
[kernel] PageFault in application, bad addr = 0x0, bad instruction = 0x804003a4, kernel killed it.
[kernel] IllegalInstruction in application, kernel killed it.
[kernel] IllegalInstruction in application, kernel killed it.
```

```
[rustsbi] RustSBI version 0.4.0-alpha.1, adapting to RISC-V SBI v2.0.0
```

2. 深入理解 trap.S 中两个函数 __alltraps 和 __restore 的作用，并回答如下问题：

   - **L40：刚进入 __restore 时，a0 代表了什么值。请指出 __restore 的两种使用情景。**
   - **L43-L48：这几行汇编代码特殊处理了哪些寄存器？这些寄存器的的值对于进入用户态有何意义？请分别解释。**

```
ld t0, 32*8(sp)
ld t1, 33*8(sp)
ld t2, 2*8(sp)
csrw sstatus, t0
csrw sepc, t1
csrw sscratch, t2
```

   - **L50-L56：为何跳过了 x2 和 x4？**

```
ld x1, 1*8(sp)
ld x3, 3*8(sp)
.set n, 5
.rept 27
LOAD_GP %n
.set n, n+1
.endr
```

   - **L60：该指令之后，sp 和 sscratch 中的值分别有什么意义？**

```
csrrw sp, sscratch, sp
```

- **__restore：中发生状态切换在哪一条指令？为何该指令执行之后会进入用户态？**
- **L13：该指令之后，sp 和 sscratch 中的值分别有什么意义？**

```
csrrw sp, sscratch, sp
```

- **从 U 态进入 S 态是哪一条指令发生的？**

荣誉准则

1. 在完成本次实验的过程（含此前学习的过程）中，我曾分别与 以下各位 就（与本次实验相关的）以下方面做过交流，还在代码中对应的位置以注释形式记录了具体的交流对象及内容：

   无

2. 此外，我也参考了 以下资料 ，还在代码中对应的位置以注释形式记录了具体的参考来源及内容：

   无

3. 我独立完成了本次实验除以上方面之外的所有工作，包括代码与文档。 我清楚地知道，从以上方面获得的信息在一定程度上降低了实验难度，可能会影响起评分。

4. 我从未使用过他人的代码，不管是原封不动地复制，还是经过了某些等价转换。 我未曾也不会向他人（含此后各届同学）复制或公开我的实验代码，我有义务妥善保管好它们。 我提交至本实验的评测系统的代码，均无意于破坏或妨碍任何计算机系统的正常运转。 我清楚地知道，以上情况均为本课程纪律所禁止，若违反，对应的实验成绩将按“-100”分计。