实现功能：

继承get\_time,get\_task\_info,mmap和munmap

spawn：将fork与exec连在一起

Stride调度：在TCB添加priority和pass，stride，sys\_set\_priority用于设置优先级；修改TaskManager的fetch函数，遍历等待队列，获得pass最小的进程，根据pass += stride，再从队列中弹出这个进程。BIG\_STRIDE取1e5

问答题：

stride 算法深入

1）不是, 250 + 10 = 260 , 无符号整数溢出后回转变为 4 (260 % 256) , 4 < 255 , 所以 p2 继续执行

2）每次增大的都是最小的, 每次增大的步幅小的进程调度的次数多, 每次增大的步幅大的进程调度的次数少 {% asset\_img stride.png stride调度进程PASS值变化 %}

3）代码如下

use core::cmp::Ordering;

struct Pass(u64);

impl PartialOrd for Pass {

fn partial\_cmp(&self, other: &Self) -> Option<Ordering> {

let cmp: i64 = (self.0 - other.0) as i64;

if cmp < 0 {

Some(Ordering::Less)

} else if cmp == 0 {

Some(Ordering::Equal)

} else {

Some(Ordering::Greater)

}

}

}

impl PartialEq for Pass {

fn eq(&self, other: &Self) -> bool {

false

}

}