

增加新功能：

- 当vi一个不存在的文件时可以直接新建一个文件：

```
/root> vi a
File a not found.
Dou you want to create a new File called:a?
Y/N:
N
/root> vi a
File a not found.
Dou you want to create a new File called:a?
Y/N:
Y
File a created with size 1B.
/root> ls
a
```

- 增加抢占优先级调度：

在程序开始时可进行选择：

```
请选择进程调度算法：
1. FCFS
2. SJF
3. RR
4. PRIORITY
5. MLFQ（4个队列，优先级从高到低，时间片逐渐增多）
6. PRIORITY_Preemptive
输入数字选择： 6
调度策略已设置为： PRIORITY_Preemptive
CPU-0 已注册到调度器
CPU-1 已注册到调度器
CPU-2 已注册到调度器
CPU-3 已注册到调度器
```

当某进程在运行时，ready队列中有一个优先级高于其的进程，原进程会保存当前指令的运行状态（实际上，其他指令要么一瞬间完成，要么执行时会挂起（如IO），只有C会被中断）

实现逻辑：进程每次运行单元时间片后会触发时钟中断，查看队列中是否存在优先级更高的进程，若有则替换并保存当前进程的运行状态。

如下图为申请了4个CPU，启动了8个进程，**已验证正确性：**

```
/root> exec a 8 a 7 a 6 a 5 a 4 a 3 a 2 a 1
进程 1 加入优先级为 0 的就绪队列
调度器分配进程 1 到 CPU-0
CPU-0 开始执行进程 1
创建进程 1，执行文件：a，优先级为 8
进程 2 加入优先级为 0 的就绪队列
调度器分配进程 2 到 CPU-1
CPU-1 开始执行进程 2
创建进程 2，执行文件：a，优先级为 7
进程 3 加入优先级为 0 的就绪队列
调度器分配进程 3 到 CPU-2
CPU-2 开始执行进程 3
创建进程 3，执行文件：a，优先级为 6
进程 4 加入优先级为 0 的就绪队列
调度器分配进程 4 到 CPU-3
CPU-3 开始执行进程 4
创建进程 4，执行文件：a，优先级为 5
```

```
C 1000
CPU-3 执行指令：C 1000
CPU-0 执行指令：C 1000
CPU-1 执行指令：C 1000
CPU-2 执行指令：C 1000
CPU-2 进程 3被抢占 已使用时间片 100/1000
进程 3 加入优先级为 0 的就绪队列
CPU-2 开始执行进程 6
C 1000
CPU-2 执行指令：C 1000
CPU-1 进程 2被抢占 已使用时间片 100/1000
进程 2 加入优先级为 0 的就绪队列
CPU-1 开始执行进程 7
C 1000
CPU-1 执行指令：C 1000
CPU-0 进程 1被抢占 已使用时间片 100/1000
进程 1 加入优先级为 0 的就绪队列
CPU-0 开始执行进程 5
CPU-3 进程 4被抢占 已使用时间片 100/1000
进程 4 加入优先级为 0 的就绪队列
CPU-3 开始执行进程 8
C 1000
CPU-3 执行指令：C 1000
C 1000
CPU-0 执行指令：C 1000
CPU-1 执行指令：C 900
CPU-2 执行指令：C 900
```

修改及完善：

修改了原程序中多级反馈调度逻辑上的一些错误，
并将第四个队列由FCFS改为了RR（timeslice=800）

存在的问题：

现在中断处理的函数直接在cpu中写了，而中断包中的类及方法几乎没用，并且有重复，后面需要降低耦合性，在彻底完善后将中断部分从cpu中拆开。