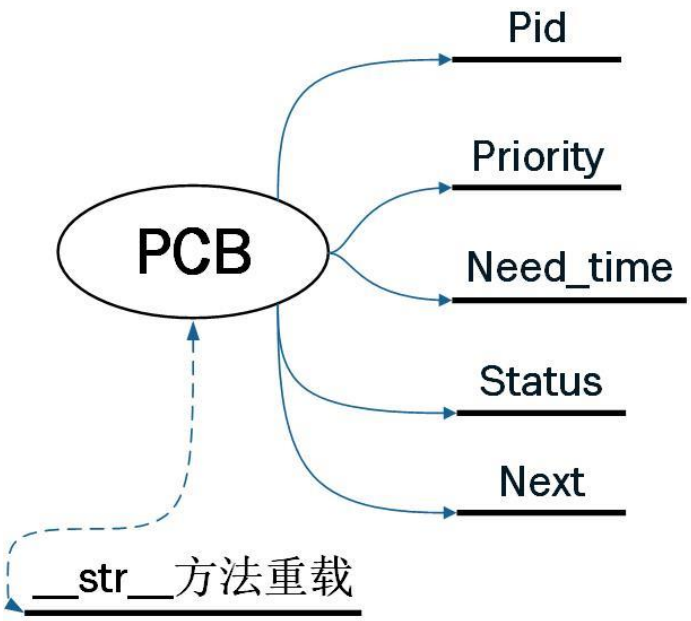
OS 静态优先级抢占算法实验报告

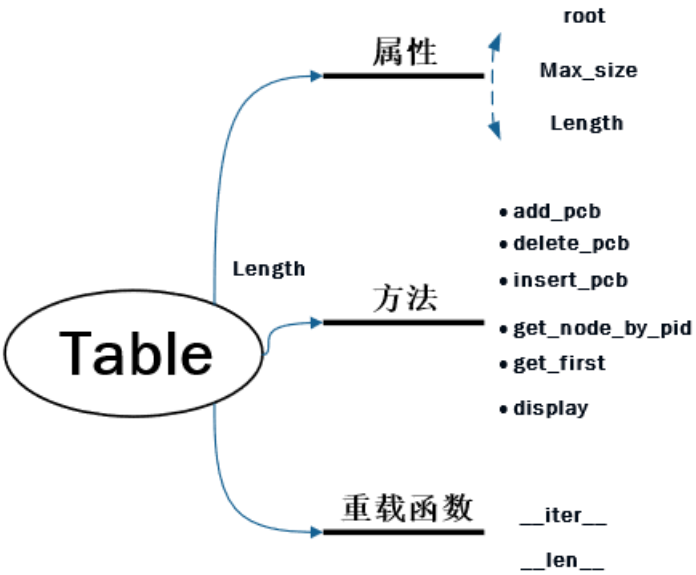
1 PCB 基本结构介绍



2 PCB 储存结构

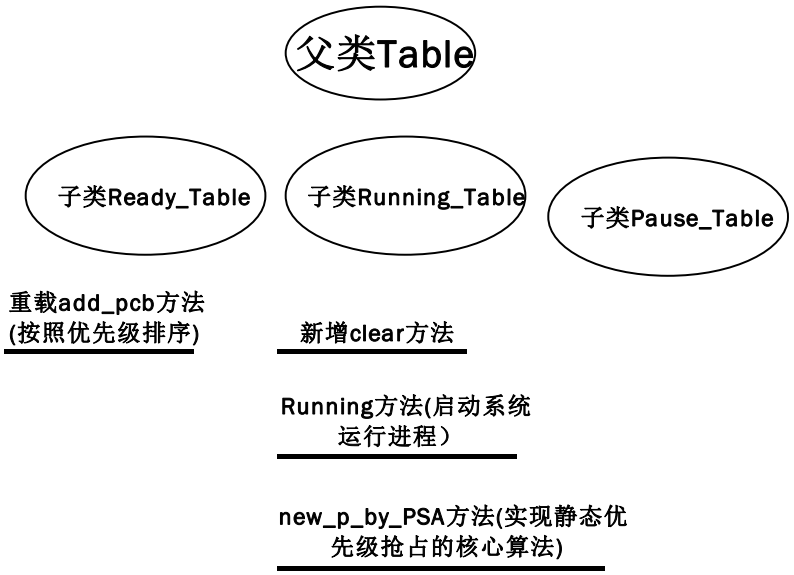
2.1 基本 class：table

Table 是自定的一个 class，其主要结构是单链表，同时该 class 封装了对 table 进行增删改查的所有方法。另外重载了该类的__iter__方法，使该类成为了一个可迭代对象；定义了一个 display 方法，用于打印输出 table 存储的所有 pcb 节点，具体结构见下：



2.3 子 类 class:

- Ready_table(用于存放就绪状态的 pcb)
- Pause_table(用于存储挂起状态的 pcb)
- Runnig_table(用于存储当前正在运行的 pcb)



3 核心算法

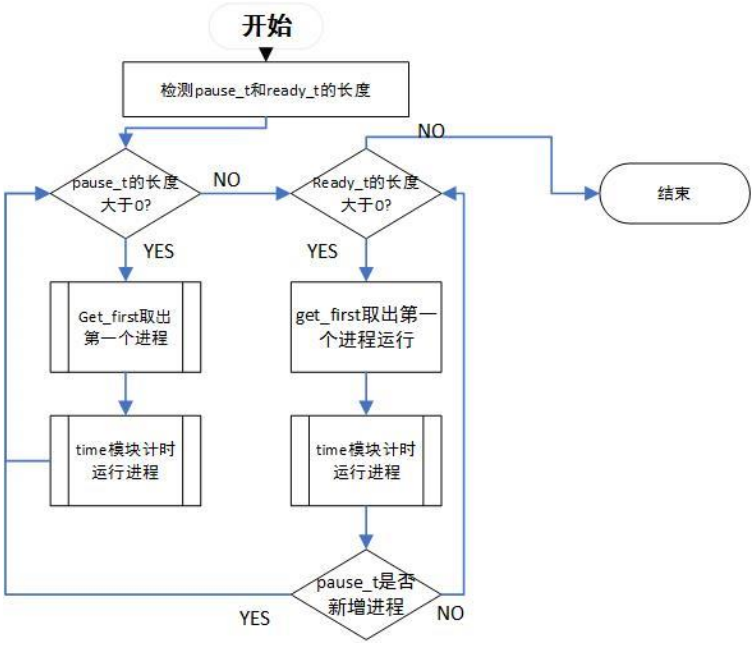
3.1 Running 算法

如果 pause_t 里面有就绪的进程，则调用 get_first 方法取出 pcb，加载到 running_t 里面进行运行。

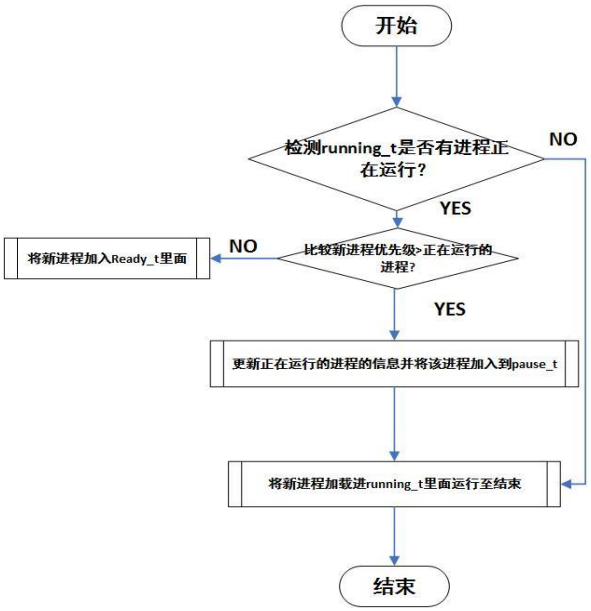
如果 pause_t 里面没有进程，则运行 ready_t 里面就绪的进程。

同时，在运行过程中，也要不断检测 pause_t 里面是否新增了进程。

如果新增了，则中断去取出 pause_t 的进程运行。

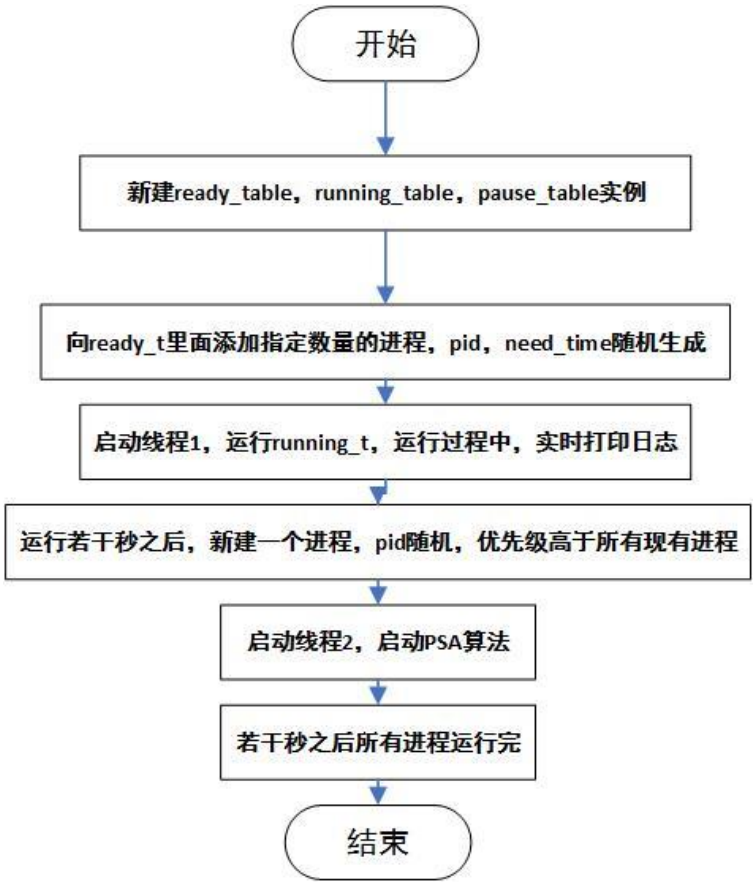


3.2 静态优先级抢占算法(new_p_by_PSA)



4 系统启动

Test 函数思路如下



函数中使用了 threading 模块，使得 running 函数和 PSA 函数能够并行运行
相应代码为：

```
t2.start()
t1.join()
t2.join()
```

5 运行效果

测试日志如下:

```
------------------------
pid:2,status:ready,priority:1
is inserted into running table by force
------------------------
pid:2,status:ready,priority:1
has been running for 1 seconds
------------------------
pid:2,status:ready,priority:1
has been running for 2 seconds
------------------------
pid:2,status:ready,priority:1
```

has been running for 3 seconds

pid:2,status:ready,priority:1---is over

---

pid:4,status:ready,priority:3

has been running for 1 seconds

---

pid:4,status:ready,priority:3

has been running for 2 seconds

---

pid:4,status:ready,priority:3

has been running for 3 seconds

---

pid:4,status:ready,priority:3

has been running for 4 seconds

pid:4,status:ready,priority:3---is over

---

pid:3,status:ready,priority:4

has been running for 1 seconds

---

pid:3,status:ready,priority:4

has been running for 2 seconds

---

pid:3,status:ready,priority:4

has been running for 3 seconds

---

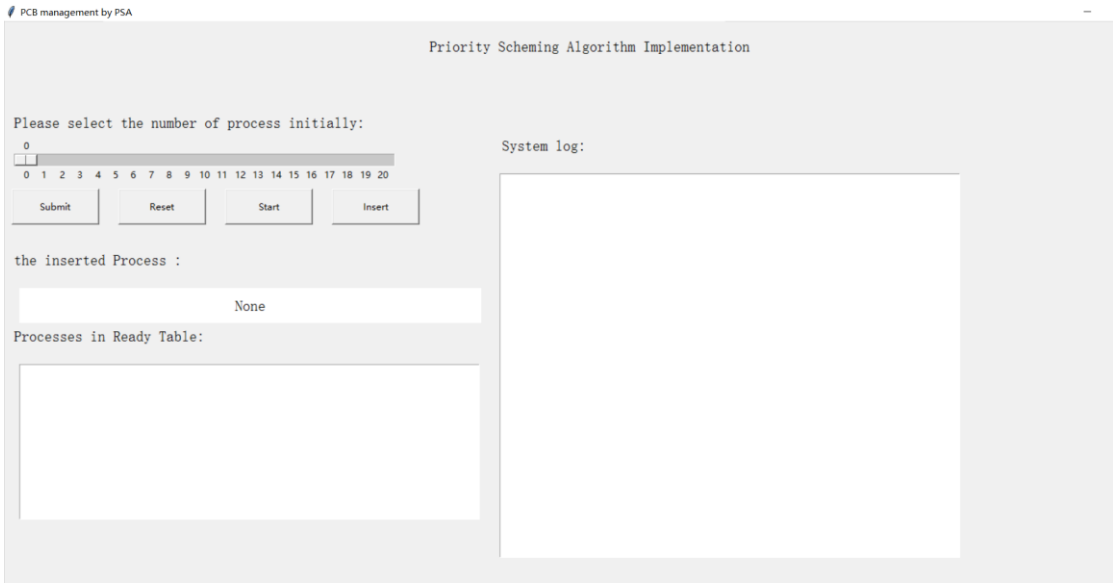
pid:3,status:ready,priority:4

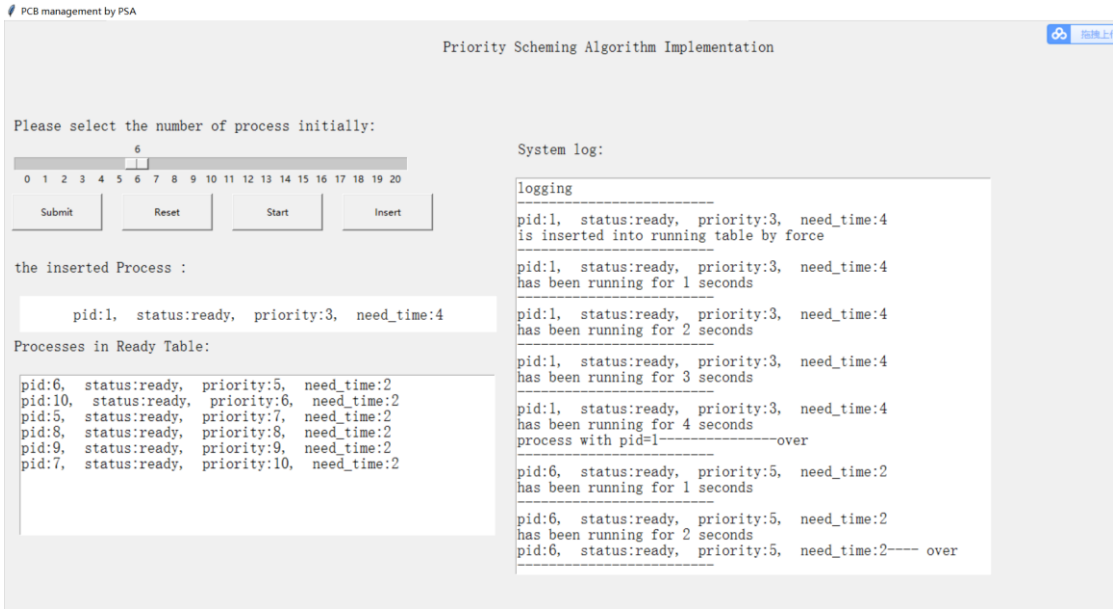
has been running for 4 seconds

pid:3,status:ready,priority:4---is over

6 GUI 设计效果

新增 GUI 界面


PCB management by PSA
Priority Scheming Algorithm Implementation
Please select the number of process initially:
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Submit
Reset
Start
Insert
System log:
the inserted Process :
None
Processes in Ready Table:


PCB management by PSA
Priority Scheming Algorithm Implementation
Please select the number of process initially:
6
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Submit
Reset
Start
Insert
System log:
the inserted Process :
pid:1, status:ready, priority:3, need_time:4
Processes in Ready Table:
pid:6, status:ready, priority:5, need_time:2
pid:10, status:ready, priority:6, need_time:2
pid:5, status:ready, priority:7, need_time:2
pid:8, status:ready, priority:8, need_time:2
pid:9, status:ready, priority:9, need_time:2
pid:7, status:ready, priority:10, need_time:2
logging
pid:1, status:ready, priority:3, need_time:4
is inserted into running table by force
pid:1, status:ready, priority:3, need_time:4
has been running for 1 seconds
pid:1, status:ready, priority:3, need_time:4
has been running for 2 seconds
pid:1, status:ready, priority:3, need_time:4
has been running for 3 seconds
pid:1, status:ready, priority:3, need_time:4
has been running for 4 seconds
process with pid=1----------over
pid:6, status:ready, priority:5, need_time:2
has been running for 1 seconds
pid:6, status:ready, priority:5, need_time:2
has been running for 2 seconds
pid:6, status:ready, priority:5, need_time:2---- over