

lab6 实验文档

孔浩宇

一、实验目标

1. 本实验在自己所完成的实验5的基础上进行，提供示例框架，但不要求必须使用示例框架
2. 调度算法，至少2种（不含FCFS，必须含有抢占式调度算法）
3. 示例框架给出了测试样例，也可自己修改实现
4. Hook 机制的实现（示例框架中 scheduler.h 与 scheduler.c 已实现）

```
typedef struct scheduler {  
    unsigned long type;    // the type of the scheduler  
    int preemptive_or_not; //if True, the scheduler is preemptive  
    myTCB* (*nextTsk_func)(void);  
    void (*enqueueTsk_func)(myTCB *tsk);  
    void (*dequeueTsk_func)(myTCB *tsk);  
    void (*schedulerInit_func)(myTCB* idleTsk);  
    int (*createTsk_hook)(void (*tskBody)(void),tskPara para);  
    void (*tick_hook)(void);    //if set, tick_hook will be called every tick  
}scheduler;
```

二、实验内容

1. TCB 队列维护及函数接口

示例框架中三个调度算法使用同一个队列，只在入队时采用不同的函数，控制其为先进先出/优先级排序。

示例框架中需要补全的函数在 `task.c` 中使用 `TODO` 标注。

```
typedef struct rdyQueueFCFS {
    myTCB* head;
    myTCB* tail;
    myTCB * idleTask;
} rdyQueueFCFS;
rdyQueueFCFS rqFCFS;
void rqFCFSInit();
void initT_RR();
int rqEmpty_FCFS();
void taskEnqueue_FCFS(struct myTCB *task);
void taskEnqueue_PRIO(struct myTCB *task);
void taskDequeue_FCFS();
void schedule_FCFS(void);
void schedule_RR(void);
myTCB* nextTask_FCFS();
myTCB* nextTask_RR();
```

2.任务参数模块

对于myTCB新增任务参数模块，以及相关函数。对应函数作用在示例框架中均已标出

```
typedef struct taskPara
{
    unsigned int priority;
    unsigned int arrTime;
    unsigned int exetime;
} taskPara;

typedef struct myTCB
{
    unsigned long *stackTop;
    unsigned long TSK_ID;
    unsigned long TSK_State;
    unsigned long runTime;
    unsigned long thisRunTime;
    unsigned long *stackMax;
    void (*task_entrance)(void);
    taskPara* taskPara;
    struct myTCB *nextTCB;
} myTCB;

void initTaskPara(struct taskPara*buffer);
void setTaskPara(unsigned int tag,unsigned int value,struct taskPara* buffer);
unsigned int getTaskPara(int tag,struct taskPara *para);
```

3. 到达队列的维护

到达队列的实现，基本与PRIO的队列接口一致，添加一个定期将任务添加到执行队列的函数并hook到时钟上。

```
extern int arrvLength;
struct arrvNode
{
    struct myTCB* TCB;
    struct arrvNode* next;
};

struct arrv
{
    struct arrvNode* head;
    struct arrvNode* tail;
};

extern struct arrv arrvData;
void addTask(int tid);
void arrvHook();
void arrvInit();
void arrvEnqueue(struct myTCB* task);
int arrvEmpty();
void arrvDequeue();
```

4. 调度器接口与实现

具体在 scheduler.c 中实现，注意函数接口的复用，即不同的调度算法，哪些操作是一致的。

三、实验测试

1.测试样例

示例框架中的测试样例如下，也可自行实现。

```
void TEST(void)
{

    int newTskTid1 = createTask(myTskFCFS1); // its tid will be 1
    setTaskPara(PRIO, 3, tcbPool[newTskTid1].taskPara);
    setTaskPara(ARRTIME, 0, tcbPool[newTskTid1].taskPara);
    setTaskPara(EXETIME, 3, tcbPool[newTskTid1].taskPara);

    int newTskTid2 = createTask(myTskFCFS2); // its tid will be 2
    setTaskPara(PRIO, 1, tcbPool[newTskTid2].taskPara);
    setTaskPara(ARRTIME, 1, tcbPool[newTskTid2].taskPara);
    setTaskPara(EXETIME, 2, tcbPool[newTskTid2].taskPara);

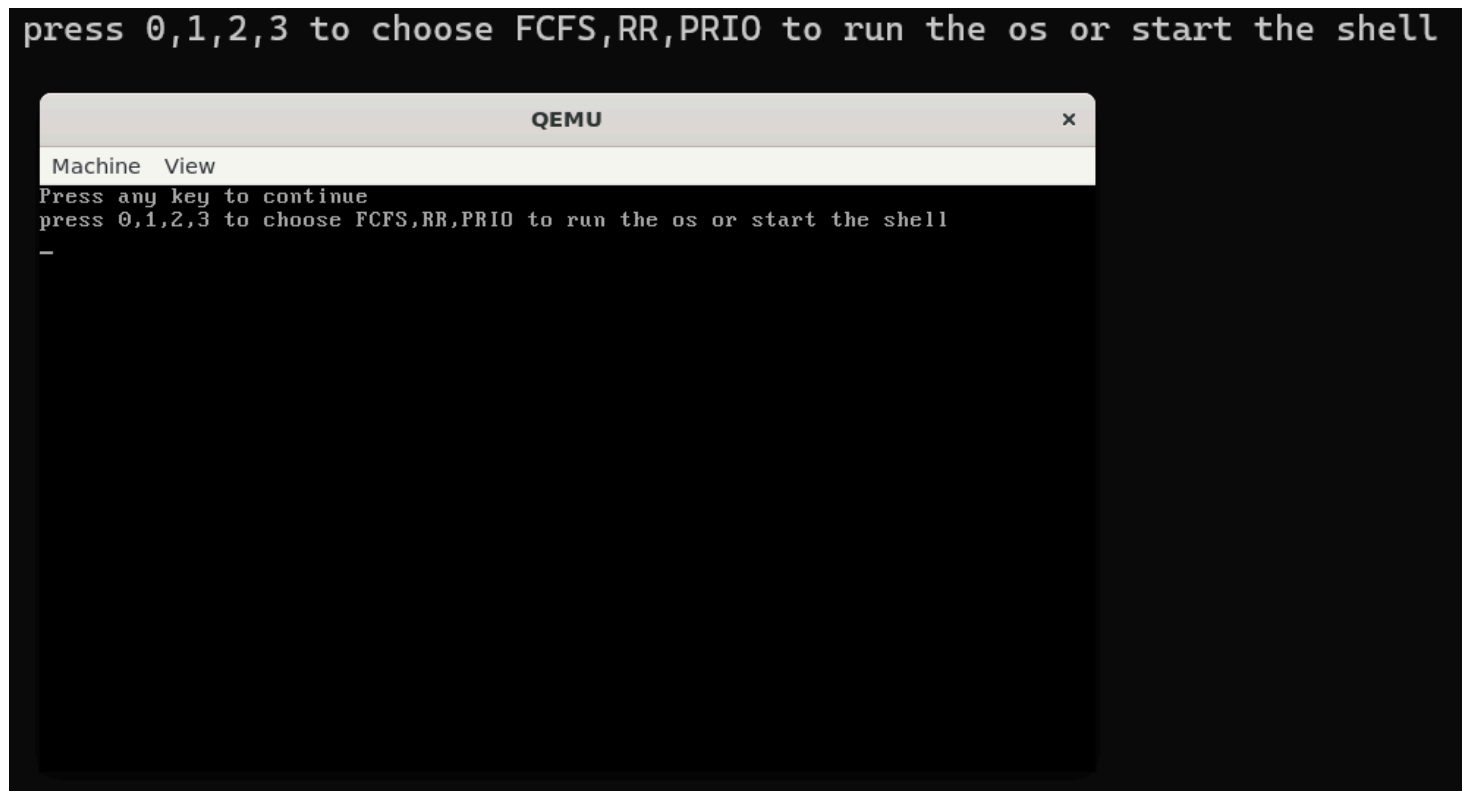
    int newTskTid3 = createTask(myTskFCFS3); // its tid will be 3
    setTaskPara(PRIO, 0, tcbPool[newTskTid3].taskPara);
    setTaskPara(ARRTIME, 3, tcbPool[newTskTid3].taskPara);
    setTaskPara(EXETIME, 3, tcbPool[newTskTid3].taskPara);

    int newTskTid4 = createTask(myTskFCFS4); // its tid will be 4
    setTaskPara(PRIO, 4, tcbPool[newTskTid4].taskPara);
    setTaskPara(ARRTIME, 2, tcbPool[newTskTid4].taskPara);
    setTaskPara(EXETIME, 3, tcbPool[newTskTid4].taskPara);

    addTask(newTskTid1);
    addTask(newTskTid2);
    addTask(newTskTid3);
    addTask(newTskTid4);
}
```

2.测试结果

初始界面



FCFS 调度

根据先到先服务原则，执行顺序应为1→2→4→3

```
START RUNNING.....
*****
*          INIT   INIT !          *
*****
*****
* current task tid : 1             *
* priority          : 3             *
* arrive time       : 0             *
* execute time      : 1 / 3         *
*****
*****
* current task tid : 1             *
* priority          : 3             *
* arrive time       : 0             *
* execute time      : 2 / 3         *
*****
*****
* current task tid : 1             *
* priority          : 3             *
* arrive time       : 0             *
* execute time      : 3 / 3         *
*****
*****
* current task tid : 2             *
* priority          : 1             *
* arrive time       : 1             *
* execute time      : 1 / 2         *
*****
*****
* current task tid : 2             *
* priority          : 1             *
* arrive time       : 1             *
* execute time      : 2 / 2         *
*****
*****
* current task tid : 4             *
* priority          : 4             *
* arrive time       : 2             *
* execute time      : 1 / 3         *
*****
*****
* current task tid : 4             *
* priority          : 4             *
* arrive time       : 2             *
* execute time      : 2 / 3         *
*****
*****
```

QEMU

Machine View

```
* current task tid : 2             *
* priority          : 1             *
* arrive time       : 1             *
* execute time      : 1 / 2         *
*****
*****
* current task tid : 2             *
* priority          : 1             *
* arrive time       : 1             *
* execute time      : 2 / 2         *
*****
*****
* current task tid : 4             *
* priority          : 4             *
* arrive time       : 2             *
* execute time      : 1 / 3         *
*****
*****
* current task tid : 4             *
* priority          : 4             *
* arrive time       : 2             *
* execute time      : 2 / 3         *
*****
*****
```

1919:00:06

```
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 1 / 3  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 2 / 3  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 3 / 3  *
*****
*****
* current task tid : 3      *
* priority          : 0      *
* arrive time       : 3      *
* execute time      : 1 / 3  *
*****
*****
* current task tid : 3      *
* priority          : 0      *
* arrive time       : 3      *
* execute time      : 2 / 3  *
*****
*****
* current task tid : 3      *
* priority          : 0      *
* arrive time       : 3      *
* execute time      : 3 / 3  *
*****
*****
```

QEMU

Machine View

```
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 3 / 3  *
*****
*****
* current task tid : 3      *
* priority          : 0      *
* arrive time       : 3      *
* execute time      : 1 / 3  *
*****
*****
* current task tid : 3      *
* priority          : 0      *
* arrive time       : 3      *
* execute time      : 2 / 3  *
*****
*****
* current task tid : 3      *
* priority          : 0      *
* arrive time       : 3      *
* execute time      : 3 / 3  *
*****
*****
```

19:00:18

RR 调度

根据时间片轮转调度，任务在执行2个时间后到队尾，此时队列顺序241。

- 执行任务2，任务2执行2个时间结束，此时队列顺序413。
- 执行任务4，任务4执行2个时间后到队尾，此时队列顺序134。
- 执行任务1，1个时间后结束。再执行任务3，执行2个时间后到队尾，此时队列顺序43。
- 执行任务4，1个时间后结束，再执行任务3，一个时间后结束。

```
press 0,1,2,3 to choose FCFS,RR,PRIQ to run the os or start the shell
1
MemStart: 100000
MemSize: 7f00000
_end: 107c30
START RUNNING.....
*****
*      INIT      INIT !      *
*****
*****
* current task tid : 1      *
* priority          : 3      *
* arrive time       : 0      *
* execute time      : 1 / 3  *
*****
*****
* current task tid : 1      *
* priority          : 3      *
* arrive time       : 0      *
* execute time      : 2 / 3  *
*****
*****
* current task tid : 2      *
* priority          : 1      *
* arrive time       : 1      *
* execute time      : 1 / 2  *
*****
*****
* current task tid : 2      *
* priority          : 1      *
* arrive time       : 1      *
* execute time      : 2 / 2  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 1 / 3  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 2 / 3  *
*****
*****
```



```
*****
* current task tid : 4          *
* priority          : 4          *
* arrive time       : 2          *
* execute time      : 1 / 3     *
*****
*****
* current task tid : 4          *
* priority          : 4          *
* arrive time       : 2          *
* execute time      : 2 / 3     *
*****
*****
* current task tid : 1          *
* priority          : 3          *
* arrive time       : 0          *
* execute time      : 3 / 3     *
*****
*****
* current task tid : 3          *
* priority          : 0          *
* arrive time       : 3          *
* execute time      : 1 / 3     *
*****
*****
* current task tid : 3          *
* priority          : 0          *
* arrive time       : 3          *
* execute time      : 2 / 3     *
*****
*****
* current task tid : 4          *
* priority          : 4          *
* arrive time       : 2          *
* execute time      : 3 / 3     *
*****
*****
* current task tid : 3          *
* priority          : 0          *
* arrive time       : 3          *
* execute time      : 3 / 3     *
*****
*****
```

QEMU

Machine View

```
* current task tid : 3          *
* priority          : 0          *
* arrive time       : 3          *
* execute time      : 1 / 3     *
*****
*****
* current task tid : 3          *
* priority          : 0          *
* arrive time       : 3          *
* execute time      : 2 / 3     *
*****
*****
* current task tid : 4          *
* priority          : 4          *
* arrive time       : 2          *
* execute time      : 3 / 3     *
*****
*****
* current task tid : 3          *
* priority          : 0          *
* arrive time       : 3          *
* execute time      : 3 / 3     *
*****
*****
Unknown interrupt1
```

19:00:12


```
*****
* current task tid : 3      *
* priority          : 0      *
* arrive time       : 3      *
* execute time      : 3 / 3  *
*****
*****
* current task tid : 2      *
* priority          : 1      *
* arrive time       : 1      *
* execute time      : 1 / 2  *
*****
*****
* current task tid : 2      *
* priority          : 1      *
* arrive time       : 1      *
* execute time      : 2 / 2  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 1 / 3  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 2 / 3  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 3 / 3  *
*****
```

QEMU x

Machine View

```
*****
* current task tid : 2      *
* priority          : 1      *
* arrive time       : 1      *
* execute time      : 2 / 2  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 1 / 3  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 2 / 3  *
*****
*****
* current task tid : 4      *
* priority          : 4      *
* arrive time       : 2      *
* execute time      : 3 / 3  *
*****
*****
```

Unknown interrupt1 19:00:38

四、提交要求

1. 说明自己的实现框架（使用示例框架可省略）
2. 若自编测试用例，详细说明测试用例的设置依据。
3. 实验结果截图，是否和预期一致。

提交格式要求不变，截止ddl为 7月3日下午11点59分.