



Project2 - HOST Dispatcher

Tutorial 2

2022年4月21日



上海交通大學
SHANGHAI JIAO TONG UNIVERSITY



1

Round Robin

2

Feedback

3

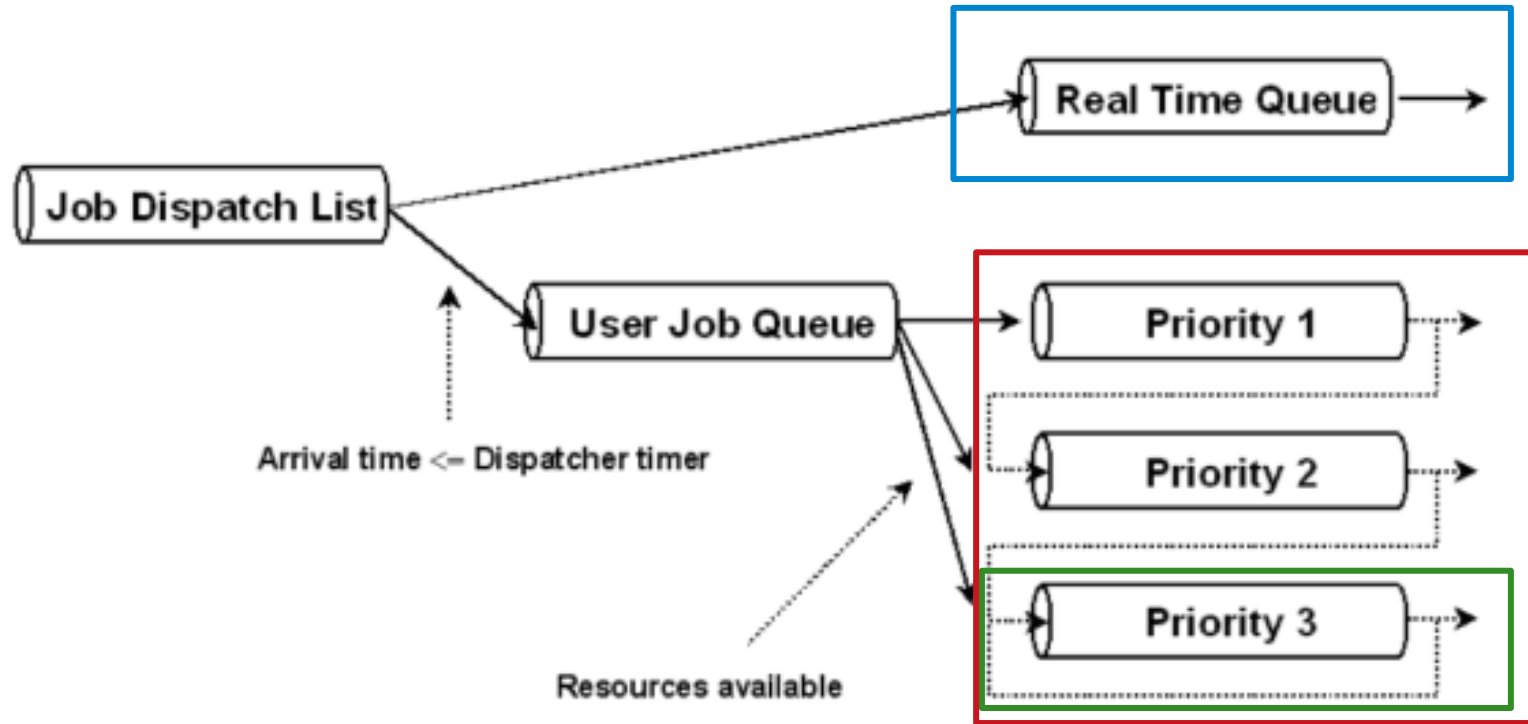
内存、IO资源

4

测试



回顾



FCFS



Round Robin

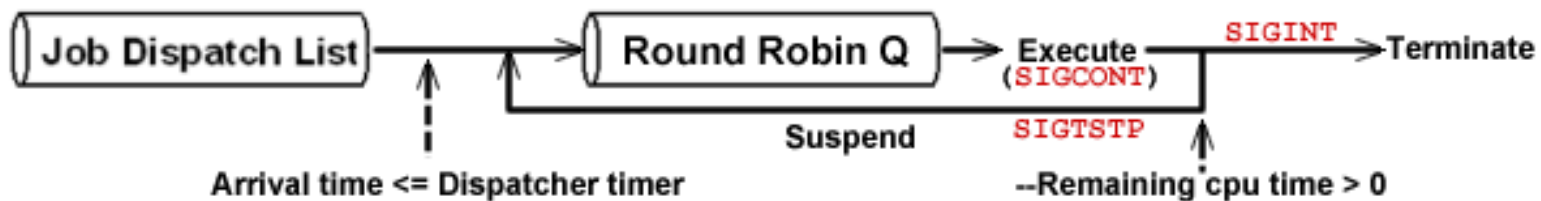


Feedback

Round Robin



- exercise08



- 信号

- SIGINT: 中断该进程。（相当于Ctrl+C）
- SIGCONT: 该进程继续运行。
- SIGTSTP: 进程暂停/挂起。

Round Robin

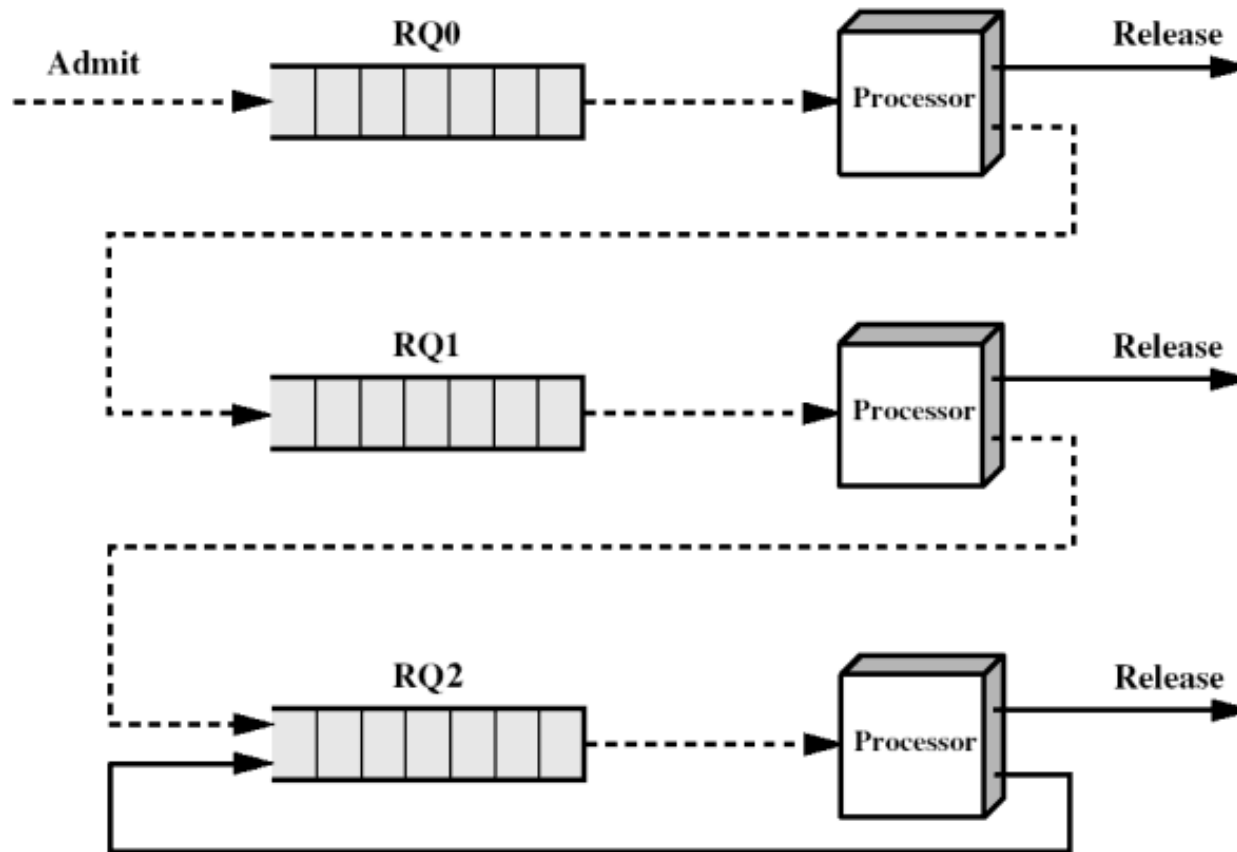


- 1.初始化调度器输入队列及RR队列
- 2-3: 初始化、读入进程（与FCFS相同）
- 4. 当任一队列非空或现在有进程在运行：
 - i) 将输入队列中已到达的进程出队，进入RR队列
 - ii) 如果当前进程非空：
 - A) 该进程的剩余cpu时间减少
 - B) 如果该进程结束，则结束该进程、释放该PCB空间
 - C) 如果未结束且RR队列中有其他进程在排队，则挂起该进程并将该进程入RR队列
 - iii) 如果当前进程为空，但RR队列非空
 - A) RR队列队首进程出队
 - B) 开始/继续该进程
 - C) 当前进程设置为该进程
 - iv) 程序sleep(1)
 - v) 计时器增加
 - vi) 返回4.
- 5. 结束

Feedback



- exercise09

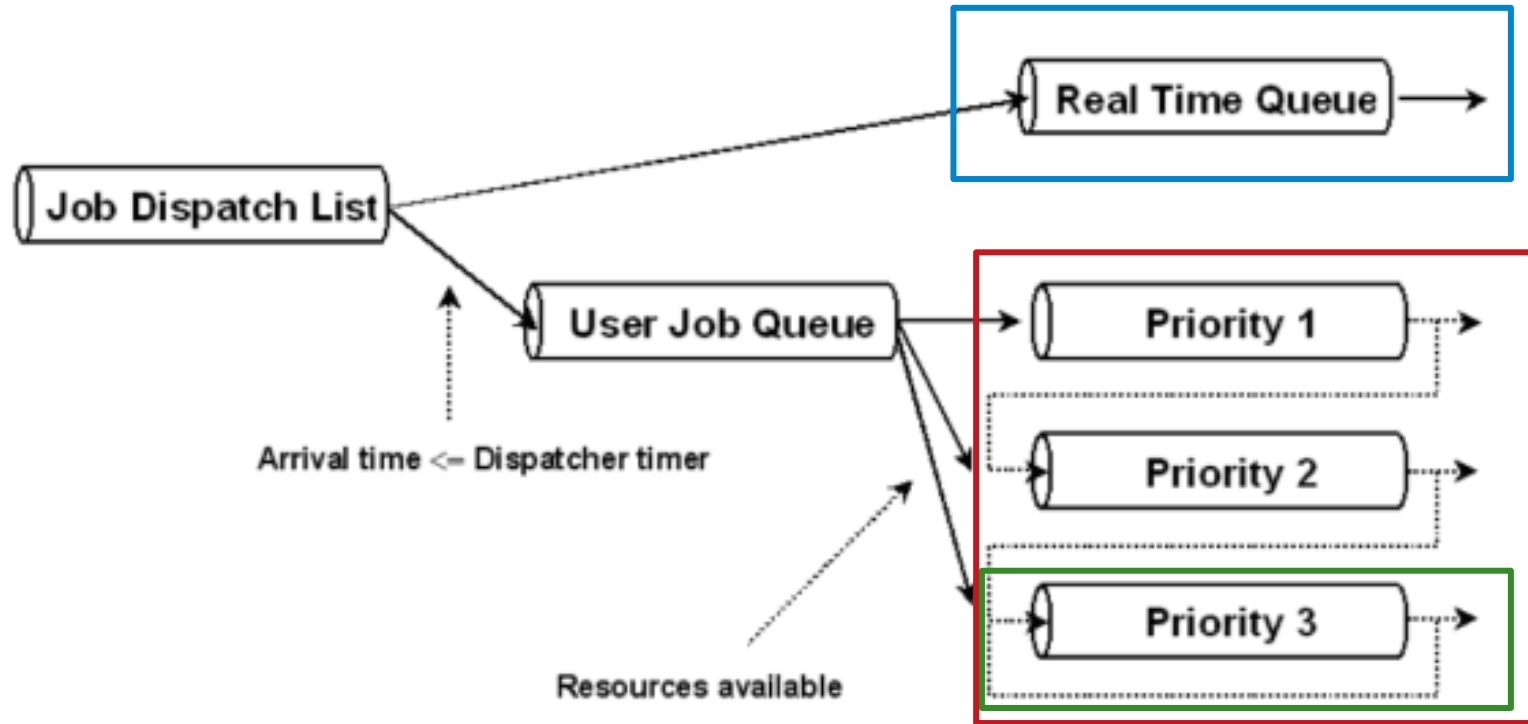


Feedback



- 1.初始化调度器**输入队列**及**feedback队列** (包含3个优先级队列, 下称fb队列)
- 2-3: 初始化、读入进程
- 4. 当**任一队列非空**或现在有进程在运行:
 - i) 将输入队列中已到达的进程出队, **并进入相应优先级队列**
 - ii) 如果当前进程非空:
 - A) 该进程的剩余cpu时间减少
 - B) 如果该进程结束, 则结束该进程、释放该PCB空间
 - C) 如果未结束且fb队列中有其他进程在排队, **则挂起该进程并将该进程优先级降低 (如需), 并放入相应优先级队列**
 - iii) 如果当前进程为空, 但fb队列非空
 - A) **从当前最高非空优先级队列中出队一个进程**
 - B) 开始/继续该进程
 - C) 当前进程设置为该进程
 - iv) 程序sleep(1)
 - v) 计时器增加
 - vi) 返回4.
- 5. 结束

HOST整体逻辑



FCFS



Round Robin



Feedback

内存-mab模块



- 一个进程的内存必须是连续的

```
52 struct mab {  
53     int offset;  
54     int size;  
55     int allocated;  
56     struct mab * next;  
57     struct mab * prev;  
58 };  
59  
60 typedef struct mab Mab;  
61 typedef Mab * MabPtr;
```

- mab.h中定义了Mab和MabPtr
 - HOST系统中的内存用双向链表进行表示
 - 链表中的每一个结点记录了一段内存信息（offset、内存大小、是否被占用）
 - 双向链表便于内存的分割和合并



内存-mab模块



- exercise10
- `MabPtr memChk(MabPtr, int);`
- `int memChkMax(int);`
- `MabPtr memAlloc(MabPtr, int);`
 - First fit, Best fit, Next fit, Worst fit 需要大家自行补全
- `MabPtr memFree(MabPtr);`
- `MabPtr memMerge(MabPtr);`
- `MabPtr memSplit(MabPtr, int);`
- `void memPrint(MabPtr);`

IO资源-rsrc模块



- 只有用户进程需要IO资源

```
35  #define MAX_PRINTERS 2
36  #define MAX_SCANNERS 1
37  #define MAX_MODEMS 1
38  #define MAX_CDS 2
39
40  struct rsrc {
41      int printers;
42      int scanners;
43      int modems;
44      int cds;
45  };
46
47  typedef struct rsrc Rsrc;
48  typedef Rsrc * RsrcPtr;
```



IO资源-rsrc模块

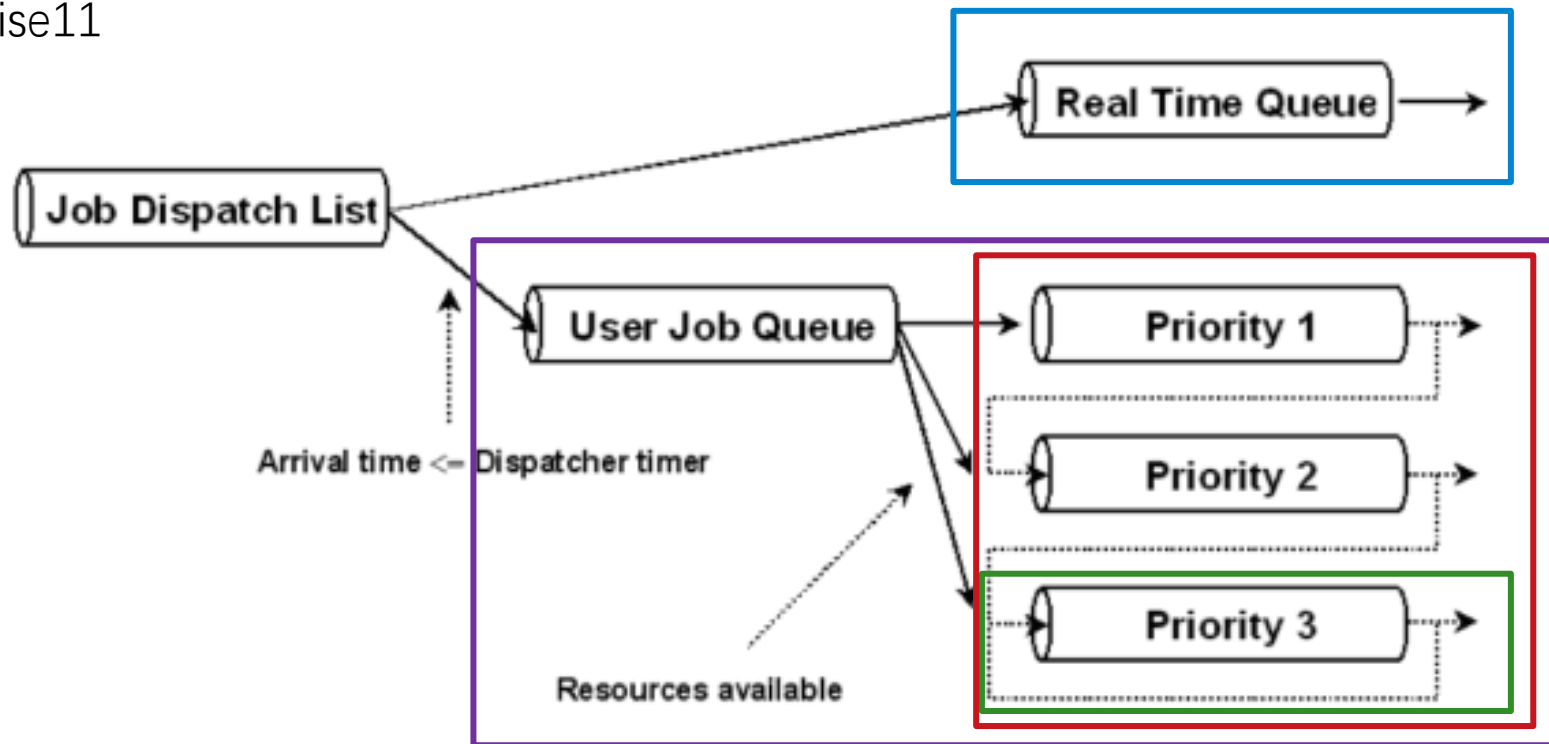


- exercise11
- `int rsrcChk(RsrcPtr available, Rsrc claim);`
- `int rsrcChkMax(Rsrc claim);`
- `int rsrcAlloc(RsrcPtr available, Rsrc claim);`
- `void rsrcFree(RsrcPtr available, Rsrc claim);`

HOST整体逻辑



exercise11



FCFS (实时进程)



Round Robin



Feedback



3-Level Feedback + resource constraint (用户进程)



测试



- testp脚本
- 提前安装tcsh工具
 - 例， ubuntu中， `sudo apt install tcsh`
- 将Canvas上Project 2–Tutorial 2下的testproject2.zip下载并解压
- 修改testp文件中的代码路径， 最好修改为绝对路径
 - 20行： testbase为你待测代码的路径（具体到exerciseXX）
 - 23行： testproject2的路径
- 在testproject2文件夹中
 - `make`
 - `./testp`
 - 即可开始测试



测试



- 测试一个工作调度表后，会生成对应.ans文件
 - 例，测试fcfs.txt后，会生成fcfs.txt.ans
- 在standard_ans/文件夹中，存放了所有测试用例的标准答案
- testp脚本会自动核对生成的.ans与标准答案的差异
 - 核对方式：Linux系统diff命令
 - 如果没有输出，则表示正确
- 也可以自行核对.ans文件的正确性

谢谢!



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

