Project2 - HOST Dispatcher

Tutorial 1

2022年4月21日

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录 Contents

Project 介绍

- HOST Dispatcher

- 四种优先级
  - 优先级0 – **实时进程**
    - 需要内存（固定为64Mb）
  - 优先级1、2、3 – **用户进程**
    - 数字越小，优先级越高
    - 用户进程遵循**三级反馈调度器**
    - 需要内存和io设备

- 工作调度表
  - 一个txt文件，描述了所有的进程

Project 介绍

- 实时进程队列
  - 用于维护从**工作调度表**中传来的**实时进程**
  - 队列中的**实时进程**必须按照FCFS算法调度
- 用户进程队列
  - 用于维护从**工作调度表**中传来的**用户进程**
  - 当队列中的用户进程所需资源（内存、io）可被满足时，才会被放入**优先级队列**中去执行
- 优先级队列
  - 与数据结构中的priority queue不一样
  - HOST中有3个**优先级队列**，对应优先级分别为1、2、3
  - 高**优先级队列**中的进程每执行一个时间片后，优先级降低1，并进入相应优先级队列重新排队
  - 3优先级队列中的进程采用Round Robin算法调度

Project 介绍

- 三种队列及HOSTD整体逻辑

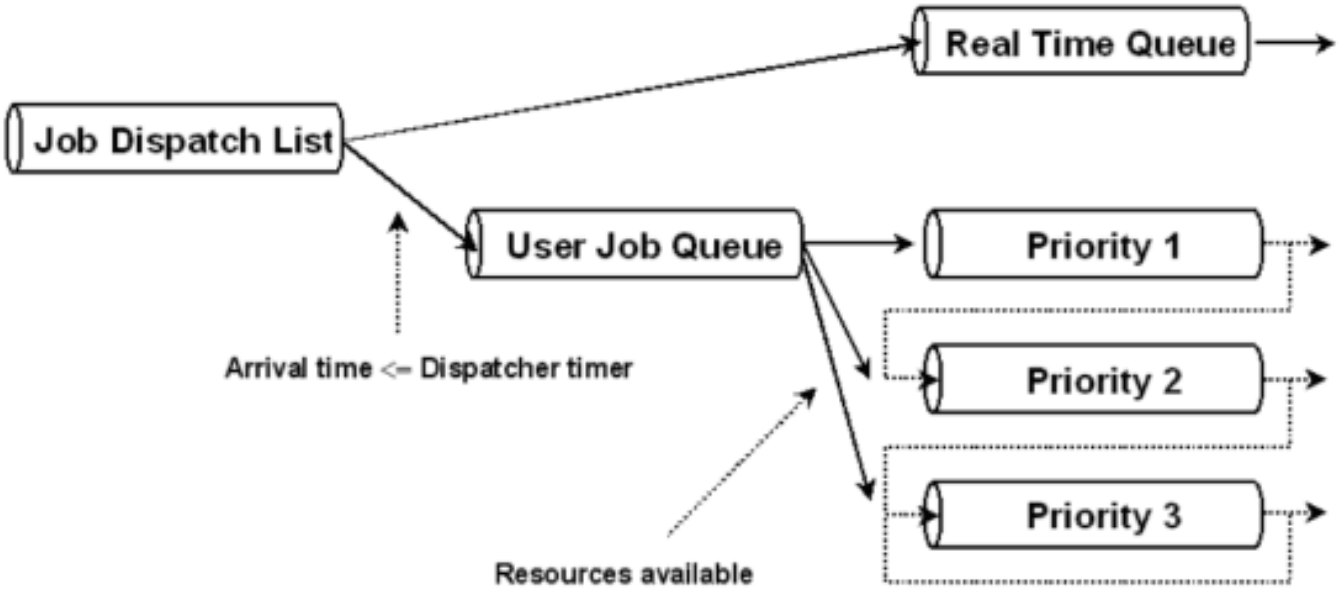


Figure 3. Dispatcher Logic Flow

Project 介绍

- 资源限制
  - IO
    - 2个打印机
    - 1个扫描仪
    - 1个调制解调器
    - 2个光驱
  - 1024Mb内存
- 内存分配
  - 一个进程分配到的内存必须是连续的
  - 实时进程的内存固定为64Mb
  - 为了保证实时进程在任意时刻都可以抢占，必须要留出至少64Mb的内存，其他960Mb内存用于用户进程

Project 介绍

- 进程
  - 源码为sigtrap.c，这部分代码请勿修改
- 工作调度表
  - 一个txt文本文件，每一行表示一个进程的信息

```
<到达时间>, <优先级>, <执行时间>, <内存大小（Mb）>, <#打印机>, <#扫描仪>, <#调制解调器>,
<#光驱>
```

- Makefile

```
make
make debug
make clean
```

代码框架

- sigtrap.c

- 一个独立的模块，编译时会生成process文件
- 模拟真实的进程（最底层）
- 本次Project不会考察大家对sigtrap.c的理解

- **请勿修改sigtrap.c中的源码！**

代码框架

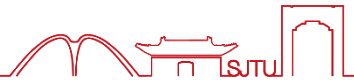
- pcb.h、pcb.c
  - 进程控制块
  - 提供了相关的API
  - 调用sigtrap.c编译成的process，用于控制该进程
- 相关API（详见pcb.c）
  - `PcbPtr startPcb(PcbPtr);`
  - `PcbPtr suspendPcb(PcbPtr);`
  - `PcbPtr terminatePcb(PcbPtr);`
  - `PcbPtr printPcb(PcbPtr, FILE *);`
  - `void printPcbHdr(FILE *);`
  - `PcbPtr createnullPcb();`
  - `PcbPtr enqPcb(PcbPtr, PcbPtr);`
  - `PcbPtr deqPcb(PcbPtr*);`

代码框架

- hostd.h、hostd.c

- HOST Dispatcher相关源代码
- 本次Project主要是对这部分代码进行编写
- 对pcb模块进行控制
  - 声明pcb队列（实时进程队列、用户进程队列、优先级队列）
  - 从工作调度表中将pcb加入队列
  - 控制pcb队列中的pcb的开始、挂起、结束
  - 检测资源是否满足
  - ……

代码框架



- mab.h、mab.c
  - 内存分配块
- rsrc.h、rsrc.c
  - IO资源管理模块
- 这部分代码本次Tutorial用不到，后面会继续介绍

FCFS

- 1. 初始化调度器**输入队列**（用于读入并暂存所有进程）
- 2. 从工作调度表读入输入队列
- 3. 计时器开始
- 4. 当输入队列非空或现在有进程在运行：
  - i) 如果有进程在运行：
    - 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
  - ii) 如果现在没有进程在运行，但输入队列非空，且输入队列的队首进程已到达
    - A) 该进程出队并开始运行
    - B) 当前进程设置为该进程
  - iii) 程序sleep(1)
  - iv) 计时器增加
  - v) 返回4.
- 5. 结束

演示

- 编译

```
make
make debug
make clean
```

- 运行

```
./hostd <dispatchlist>
```

- 平台
  - Ubuntu 18.04
  - 编译器：gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0

总结

- Project介绍
  - 两种进程、四种优先级、三种队列
  - HOST Dispatcher的整体逻辑
  - 资源（内存、IO）限制
- 代码框架
- FCFS 伪代码逻辑

谢 谢!