



中国科学技术大学

University of Science and Technology of China

011174.01: Operating System 操作系统原理与设计

Project 6: Scheduler

陈香兰(xlanchen@ustc.edu.cn)

高能效智能计算实验室, CS, USTC @ 合肥

嵌入式系统实验室, CS, USTC @ 苏州

2021/6/7

实验6基础



中国科学技术大学
University of Science and Technology of China

- 本实验在自己所完成的实验5的基础上进行



- **【必须】** 调度算法，至少2种（不含FCFS）
- **【根据调度算法需要修改】** 任务管理器
 - **【根据调度算法需要修改】** 任务数据结构
 - **【根据调度算法需要修改】** 任务创建/销毁
 - **【根据调度算法需要修改】** 调度器
- **【必须】** 自测
 - 自编测试用例

提纲



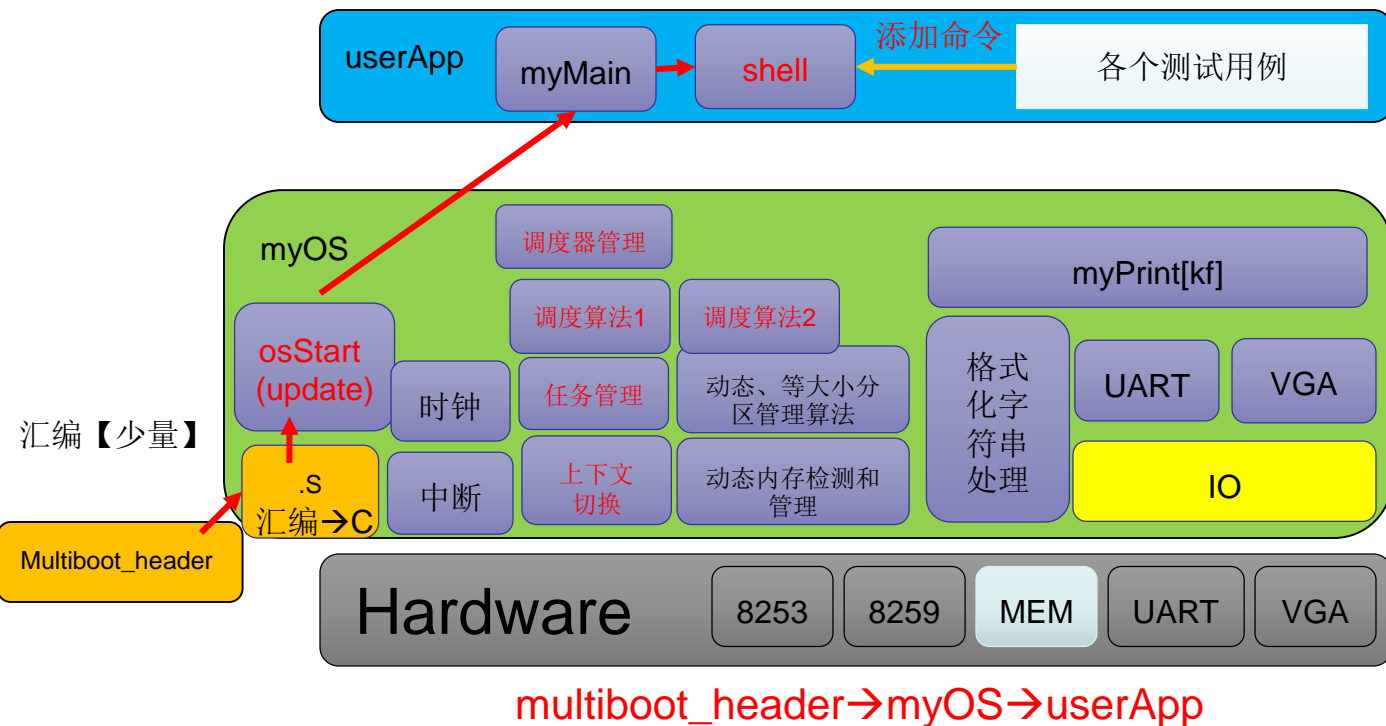
中国科学技术大学
University of Science and Technology of China

1. 软件架构和功能说明
2. 主要功能模块说明
3. 其他模块变更
4. 验收标准

1 软件架构和功能



中国科学技术大学
University of Science and Technology of China



流程:

Multiboot_header

为进入C程序准备好上下文

初始化操作系统各个模块

调用userApp入口
myMain (自测) +shell

主要功能模块【新】:

内核: 上下文切换、任务管理和调度
用户: 新功能测试

测试:

被测功能: 任务创建、所实现的调度算法
自测: userApp

主要源码文件（参考）



中国科学技术大学
University of Science and Technology of China

myOS/kernel

- |—— task.c 任务管理
- |—— taskPara.c 任务参数设置
- |—— task_arr.c 待到达任务管理
- |—— task_sched 调度算法目录，该目录下实现各种调度算法
 - |—— Makefile
 - |—— task_fifo.c
 - |—— task_fmqr.c
 - |—— task_prio.c 演示代码实现了多种调度算法
 - |—— task_prio0.c
 - |—— task_sjf.c
- |—— task_sched.c 统一的调度算法接口

...

2.1 任务管理



中国科学技术大学
University of Science and Technology of China

- myTCB中的具体内容，根据调度算法需要修改
 - 主要是修改调度相关参数
 - 可以考虑实现一种较为通用的数据结构

2.2 任务的创建和销毁



- 实现任务创建原语（接口命名为createTsk()）
 - int createTsk(void (*tskBody)(void), tskPara *para)
 - 给定任务的入口函数和任务的参数
 - 返回任务ID
- 实现任务销毁原语（接口命名为destroyTsk()）
 - void destroyTsk(int tskIndex)
 - 以任务ID为输入参数
- 任务参数tskPara 以支持设定优先级Priority/执行时长ExeTime/到达时间ArriveTime等参数

```
void initTskPara(tskPara *buffer);  
void setTskPara(unsigned int option, unsigned int value, tskPara *buffer);  
void getTskPara(unsigned option, unsigned int *para);
```

```
//option for setTskPara()/getTskPara  
#define PRIORITY      1  
#define EXETIME       2  
#define ARRTIME       3  
#define SCHED_POLICY 4
```

```
// struct for tskPara  
typedef struct tskPara{  
    unsigned int priority;  
    unsigned int exeTime;  
    unsigned int arrTime;  
    unsigned int schedPolicy;  
} tskPara;
```


2.2.1 关于任务的参数



中国科学技术大学
University of Science and Technology of China

- 优先级：用于优先级调度算法
- 到达时间：
 - 到达时间为0，表示立即就绪
 - 到达时间大于0，表示不是立即就绪
 - 需要结合周期性时钟中断，对不是立即就绪的任务进行管理，在任务到达时间释放任务
 - 注意时间单位的契合
- 执行时间：用于**SJF**调度算法
- 调度策略：用于多级队列中，不同的任务使用不同的调度策略

2.3 统一的调度相关接口



- 统一的调度接口（与具体的调度算法无关）

```
/* wrapped interfaces */  
myTCB *nextTsk(void);  
void enqueueTsk(myTCB *tsk);  
void dequeueTsk(myTCB *tsk);  
void schedulerInit();  
void createTsk_hook(myTCB *created);  
void scheduler_tick(void); // run every tick  
  
void schedule(void);
```

- 与具体的调度算法进行对接的、统一的调度器数据结构

```
struct scheduler {  
    unsigned int type;  
  
    myTCB* (*nextTsk_func)(void);  
    void (*enqueueTsk_func)(myTCB *tsk);  
    void (*dequeueTsk_func)(myTCB *tsk);  
    void (*schedulerInit_func)(void);  
    void (*createTsk_hook)(myTCB* created); //if set, will be call in createTsk (before tskStart)  
    void (*tick_hook)(void); //if set, tick_hook will be called every tick  
};
```

```
myOS/kernel/task_sched/task_fifo.c:struct scheduler scheduler_FCFS = {  
myOS/kernel/task_sched/task_fifo.c:struct scheduler scheduler_RR = {  
myOS/kernel/task_sched/task_fmQ.c:struct scheduler scheduler_FM_Q = {  
myOS/kernel/task_sched/task_prio.c:struct scheduler scheduler_PRIQ = {  
myOS/kernel/task_sched/task_prio.c:struct scheduler scheduler_MQ = {  
myOS/kernel/task_sched/task_prio0.c:struct scheduler scheduler_PRIQ0 = {  
myOS/kernel/task_sched/task_sjf.c:struct scheduler scheduler_SJF = {
```

每个调度算法都要提供这样的一个数据结构

如果学生只实现某一种调度算法，可以不采用这种方式

2.3.1 各种调度算法



- 可以实现其中的一个或者几个
 - FCFS
 - SJF
 - PRIORITY0
 - 一种特殊的优先级调度算法，每个任务的优先级各不相同
 - （每个优先级，最多只有一个任务）
 - RR
 - PRIORITY（允许多个任务为同一个优先级）
 - 同一个优先级的任务，可以选择FCFS或者RR
 - MQ：多级队列
 - FMQ：多级反馈队列

```
/*scheduler scheme*/  
#define SCHEDULER_FCFS      0  
#define SCHEDULER_SJF      1  
#define SCHEDULER_PRIORITY0 2  
#define SCHEDULER_RR       3  
#define SCHEDULER_PRIORITY 4  
#define SCHEDULER_MQ       5  
#define SCHEDULER_FMQ      6
```

2.3.2 调度器管理



中国科学技术大学
University of Science and Technology of China

- 对调度器进行算法选择、参数管理

算法选择

```
unsigned int getSysScheduler(void);  
void setSysScheduler(unsigned int what);
```

参数设定

```
void getSysSchedulerPara(unsigned int who, unsigned int *para);  
void setSysSchedulerPara(unsigned int who, unsigned int para);
```

2.8 任务管理器的初始化



中国科学技术大学
University of Science and Technology of China

- 根据需要，进行修改

- 测试用例
 - 函数级、任务级、系统级：有什么区别
- 构建系统级的测试用例
 - 生成不同的操作系统实例，每个操作系统实例采用不同的调度算法（或者调度算法使用的参数不同）
- 方法【简易】：
 - 在操作系统中，利用模块化方法，实现多种调度算法
 - 由userApp指定使用哪种调度算法（及参数配置）
 - 系统设置hook，userApp提供hook实现
 - 在任务管理器初始化的时候，根据hook来定制系统

几个测试用例



中国科学技术大学
University of Science and Technology of China

tests

- |— test0_fcfs
- |— test1_sjf
- |— test2_prio0
- |— test3_rr
- |— test3_rr1000
- |— test3_rr50
- |— test4_prio
- |— test5_mq
- |— test6_fmqs

一种参考用法:

编译时, 将某个测试用例, 复制到userApp目录中,
然后编译

换测试用例时, 只需要重新将新的测试用例复制到userApp目录下,
重新编译

4 验收标准



中国科学技术大学
University of Science and Technology of China

- 提交：源代码打包 + 实验报告；验收标准如下：
 - 完成源代码编写和调试，能够编译正确运行
 - 实现主流程，提供规定接口
 - 实现主要功能，提供规定接口
 - 将源代码进行合理的组织、提供相应的Makefile，能够生成myOS
 - 提供编译和运行脚本
 - 提交实验报告，实验报告中包括
 - 给出软件的框图，并加以概述
 - 详细说明主流程及其实现，画出流程图
 - 详细说明主要功能模块及其实现，画出流程图
 - 源代码说明（目录组织、Makefile组织）
 - 代码布局说明（地址空间）
 - 编译过程说明
 - 运行和运行结果说明
 - 遇到的问题和解决方案说明

演示



中国科学技术大学
University of Science and Technology of China

Q & A