

增加内容

- SYS_setpriority仅仅做返回，不修改优先级
- loader.c的run_all_app的"user_shell"改成"ch5_usershell"
- 添加一个新的系统调用SYS_spawn，使用fork+exec代码实现，删去了不必要的页表复制以及页表清理

问答题

1、使用vfork + copy on write，修改时才进行复制

2、fork它：变得复杂；不compose；不是安全的；不安全；速度缓慢；不scale（可伸缩性）；内存过度使用

线程安全性、复杂性、速度是我认为的比较重要的原因

3、

```
//child.c
int main() {
    int a = get_a();
    printf("a = %d", a);
    return 0;
}

//main.c
int main() {
    int a = get_a();
    if(spawn("child") == 0) {
        int b = get_b();
        printf("a + b = %d", a + b);
        exit(0);
    }
    printf("a = %d", a);
    return 0;
}
```

fork的子进程与父进程可以共享内存，因此不需要额外执行一遍get_a()

4、包括创建、就绪、等待、运行、退出这几个状态，进程首先进入创建状态，创建完毕并初始化后进入就绪态，被调度时进入运行态，运行完毕后退出，进入结束态；运行遇到等待事件时进入等待态，事件发生后再进入就绪态，运行时间片完也会进入就绪态。

fork：新建一个程序，新程序进入创建态，然后进入就绪态等待调度

exec：重新并继续执行程序，保持在运行态

wait：程序从运行态进入等待态

exit：程序从运行态进入退出态

