**操作系统课程设计实验报告**

1. 优先级调度算法：

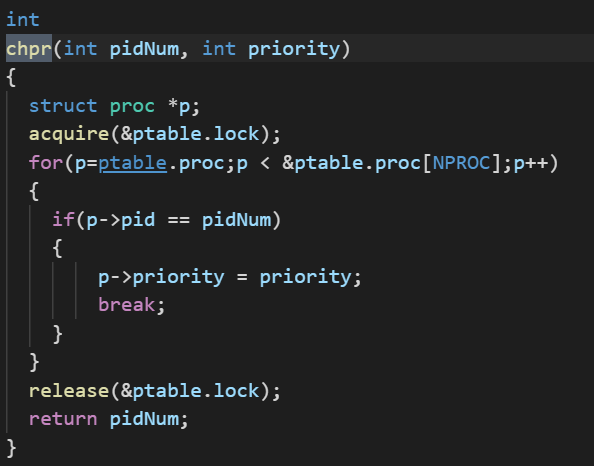


在上面的调度函数中，我们首先将最高优先级分配给0，然后在循环中再次将最高优先级分配给我们找到的第一个可运行进程，然后继续检查最高优先级。如果新的优先级高于之前的优先级，我们将最高的优先级替换为新优先级。

为了防止进程“饿死”现象，我们分别记录每个进程的运行时间、等待时间和睡眠时间。当运行时间过长，适当降低其优先级；当等待时间过长，适当提高其优先级。

1. 修改过的文件：
2. proc.c：

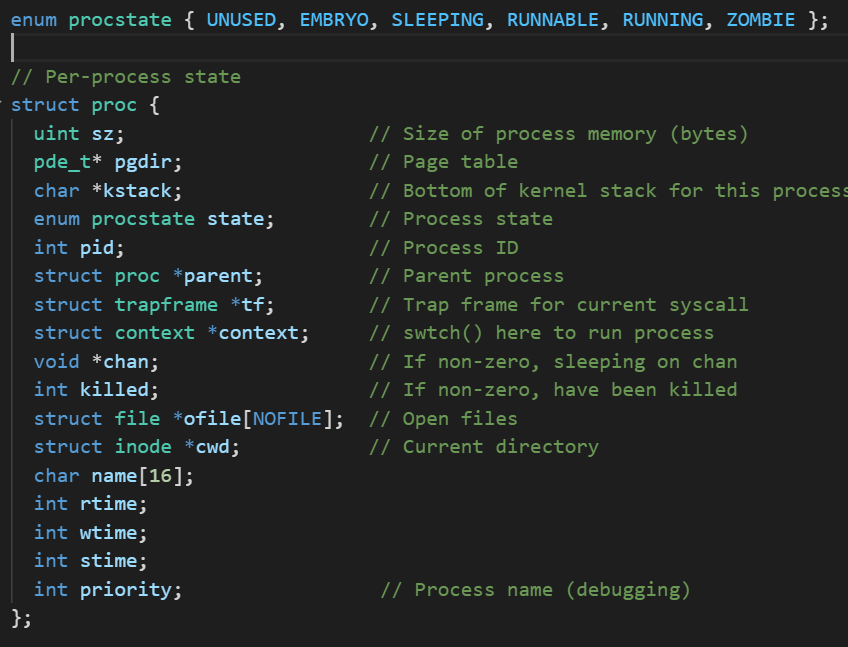
我们添加chpr函数来改变进程的优先级。



修改scheduler函数（见图一）实现优先级调度。

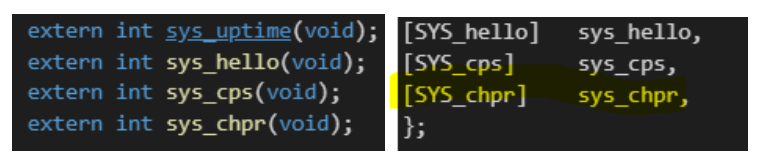
1. proc.h：

为进程结构添加属性，用来记录每个进程的运行时间、等待时间和睡眠时间。



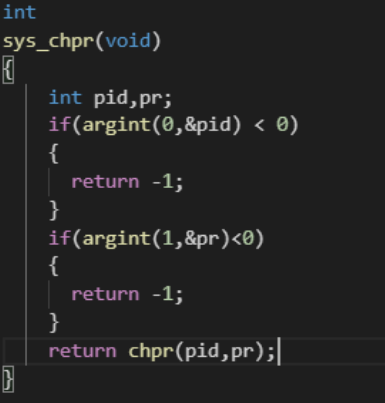
1. syscall.c:：

添加相应的系统调用。



1. sysproc.c：

添加sys\_chpr修改系统优先级。



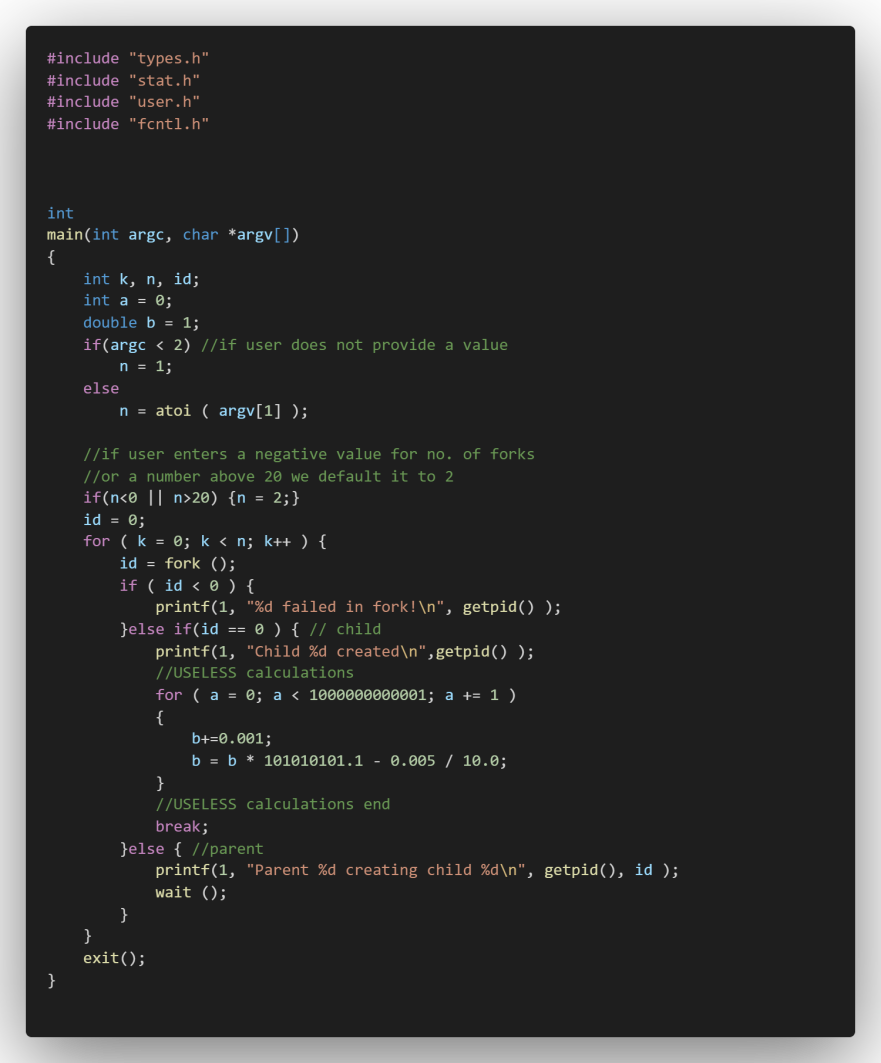
1. cpr.c：

用来编译成用户应用。



1. myfork.c：

用来创建示例进程。



1. 实验结果：

