实验九 处理机调度

1、 实验目的

加深对操作系统调度机制的理解；

2、 实验准备

拷贝 lab9 中文件（课堂上提供）到~/raspberry/prj_os/lab9，

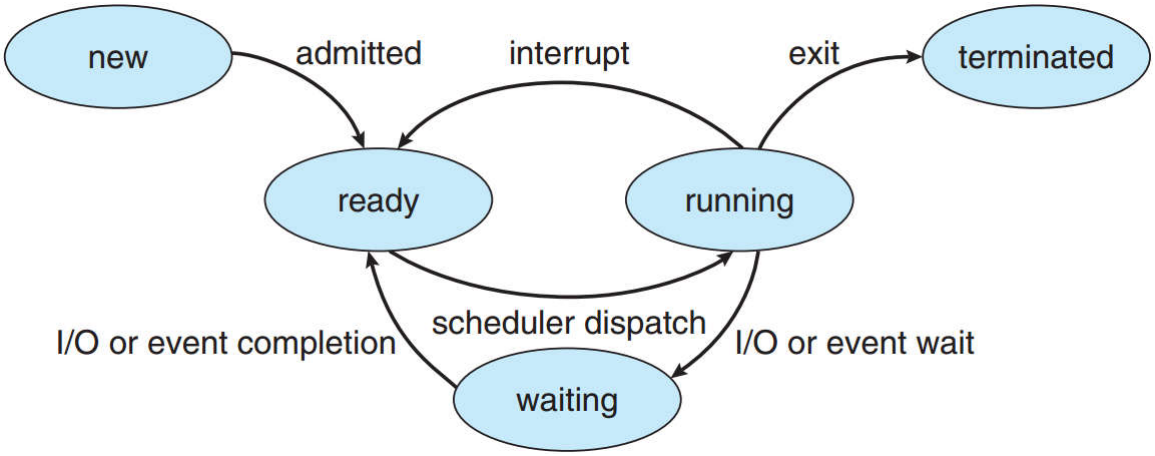
执行命令

make

编译；

3、 实验内容

在采用多道系统的程序设计中，往往有若干进程同时处于就绪状态。当就绪状态进程数大于处理机数时，就必须按照某种策略来决定哪些进程优先占用处理机。本实验模拟在单处理机情况下处理机调度。



3.1 按时间片轮转法实现处理机调度

3.1.1 设计思路

1) 每个进程用一个进程控制块 PCB 来代表，进程控制块包括进程名(进程的标识)、已运行时间、要求运行时间、状态(运行、就绪、结束) 、PCB 指针(把进程连成队列，用指针指出队列中下一个进程控制块首地址)；
2) 队列分就绪队列和完成队列；
3) 每模拟运行一次进程，已运行时间加一，要求运行时间减一；
4) 进程运行一次后，若进程要求运行时间未全部用完（不为 0），表示进程未执行完，将该进程插入到就绪队列队尾；若进程要求运行时间全部用完（为 0），则表示进程完成，进程状态改为结束，退出就绪队列，插入到完成队列队首；
5) 若就绪队列不为空，重复步骤 3）和 4）；
6) 程序有显示和打印语句，每次运行后显示变化。

3.1.2 实验操作

阅读分析源代码 round-robin.c，编译并运行；

```
yangyb@ubuntu:~/raspberry/prj_os/lab9$ ./round-robin
Please enter the total number of PCB:
3
Please enter the timeround(don't be too big,1 or 2 is best):
1
Please enter the name and time of PCB:
p1 4
p2 3
p3 2
                Display is going to start:
**********************************************
  NAME   CPUTIME    NEEDTIME    COUNT    STATUS
 p1         0          4          0         R
 p2         0          3          0         W
 p3         0          2          0         W
  NAME   CPUTIME    NEEDTIME    COUNT    STATUS
 p2         0          3          0         R
 p3         0          2          0         W
 p1         1          3          0         W

  NAME   CPUTIME    NEEDTIME    COUNT    STATUS
 p3         0          2          0         R
 p1         1          3          0         W
 p2         1          2          0         W

  NAME   CPUTIME    NEEDTIME    COUNT    STATUS
 p1         1          3          0         R
 p2         1          2          0         W
 p3         1          1          0         W

  NAME   CPUTIME    NEEDTIME    COUNT    STATUS
 p2         1          2          0         R
 p3         1          1          0         W
```

```
p1          2           2           0           W

 NAME    CPUTIME     NEEDTIME     COUNT     STATUS
p3          1           1           0           R
p1          2           2           0           W
p2          2           1           0           W

 NAME    CPUTIME     NEEDTIME     COUNT     STATUS
p1          2           2           0           R
p2          2           1           0           W
p3          2           0           1           E

 NAME    CPUTIME     NEEDTIME     COUNT     STATUS
p2          2           1           0           R
p1          3           1           0           W
p3          2           0           1           E

 NAME    CPUTIME     NEEDTIME     COUNT     STATUS
p1          3           1           0           R
p2          3           0           1           E
p3          2           0           1           E

 NAME    CPUTIME     NEEDTIME     COUNT     STATUS
p1          4           0           1           E
p2          3           0           1           E
p3          2           0           1           E

yangyb@ubuntu:~/raspberry/prj_os/lab9$
```

3.2 优先调度算法实现处理机的调度

3.2.1 设计思路

1） 每个进程用一个进程控制块 PCB 来代表，进程控制块包括进程名(进程的标识)、已运行时间、要求运行时间、状态(运行、就绪、结束) 、PCB 指针(把进程连成队列，用指针指出队列中下一个进程控制块首地址)；

```
typedef struct node
```

```
{
    char name[10];          /*进程名*/
    int prio;               /*优先数*/
    int cputime;            /*占用 cpu 时间*/
    int needtime;           /*要求运行时间*/
    char state;             /*状态*/
    struct node *next;      /*指针*/
}PCB;
```

2） 队列分就绪队列（按优先数的大小排列）和完成队列；

3） 每模拟执行一次进程，优先数减一，要求运行时间减一，已运行时间加一；

4） 正在运行的进程完成一次调度运行后，若：

- 要求运行的时间=0

那么把它的状态修改为结束，且退出就绪队列，加入完成队列(队首位置)；

- 要求运行的时间>=0

如果刚完成调度的进程优先数最大，则进程队列维持不变；

如果刚完成调度的进程优先数小于其后一个进程，则重新将该进程插入到队列中（按优先数的大小插入，重置队首标志）；

作业：请将 pri-schedule.c 中的 prio()函数补充完整，实现上述 4)所描述的功能；编译运行，参照如下蓝色字体所示输入运行所需的参数，给出运行结果截图；

备注：如果有更好的想法，整个 pri-schedule.c 可以自行重新设计；

Please enter the total number of PCB:

3

Please enter the name and time and priority of PCB:

p1 4 7

p2 3 4

p3 3 6

5） 若就绪队列不为空，重复上述 3）和 4），直到所有的进程都结束；

6） 程序有显示和打印语句，每次运行后显示变化。

3.2.2 实验操作

在将 pri-schedule.c 补充完整后，编译并运行，观察是否正确模拟了优先调度算法；

4、 作业

请将 pri-schedule.c 中的 prio()函数补充完整，实现上述 4)所描述的功能；编译运行，参照如下蓝色字体所示输入运行所需的参数，给出运行结果截图（注意运行结果和题后所给出的运行结果比较，要一致）；

备注：如果有更好的想法，整个 pri-schedule.c 可以自行重新设计；

yangyb@ubuntu:~/raspberry/prj_os/lab9$
Please enter the total number of PCB:
3
Please enter the name and time and priority of PCB:
p1 4 7
p2 3 4
p3 3 6

注意比较最后运行结果和如下结果是否一致；

Display is going to start:

****

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p1	0	4	7	W
p3	0	3	6	W
p2	0	3	4	W

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p1	1	3	6	R
p3	0	3	6	W
p2	0	3	4	W

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p3	0	3	6	R
p1	2	2	5	W
p2	0	3	4	W

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p3	1	2	5	R
p1	2	2	5	W
p2	0	3	4	W

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p1	2	2	5	R
p2	0	3	4	W
p3	2	1	4	W

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p1	3	1	4	R

p2	0	3	4	W
p3	2	1	4	W

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p2	0	3	4	R
p3	2	1	4	W
p1	4	0	3	E

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p3	2	1	4	R
p2	1	2	3	W
p1	4	0	3	E

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p2	1	2	3	R
p3	3	0	3	E
p1	4	0	3	E

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p2	2	1	2	R
p3	3	0	3	E
p1	4	0	3	E

NAME	CPUTIME	NEEDTIME	PRIORITY	STATUS
p2	3	0	1	E
p3	3	0	3	E
p1	4	0	3	E