## 实验五 使用动态优先权的进程调度算法模拟

1、实验目的

通过动态优先权算法的模拟加深对进程概念进程调度过程的理解。

2、实验内容

（1）用C语言来实现对N个进程采用动态优先权优先算法的进程调度。

（2）每个用来标识进程的进程控制块PCB用结构来描述，包括以下字段：

•••• 进程标识数 ID。

•••• 进程优先数 PRIORITY，并规定优先数越大的进程，其优先权越高。

•••• 进程已占用的CPU时间CPUTIME。

•••• 进程还需占用的CPU时间ALLTIME。当进程运行完毕时，ALLTIME变为0。

•••• 进程的阻塞时间STARTBLOCK，表示当进程再运行STARTBLOCK个时间片后，将进入阻塞状态。

•••• 进程被阻塞的时间BLOCKTIME，表示已阻塞的进程再等待BLOCKTIME个时间片后，将转换成就绪状态。

•••• 进程状态STATE。

•••• 队列指针NEXT，用来将PCB排成队列。

（3）优先数改变的原则：

•••进程在就绪队列中呆一个时间片，优先数加1。

•••进程每运行一个时间片，优先数减3。

（4）假设在调度前，系统中有5个进程，它们的初始状态如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 0 | 1 | 2 | 3 | 4 |
| PRIORITY | 9 | 38 | 30 | 29 | 0 |
| CPUTIME | 0 | 0 | 0 | 0 | 0 |
| ALLTIME | 3 | 3 | 6 | 3 | 4 |
| STARTBLOCK | 2 | -1 | -1 | -1 | -1 |
| BLOCKTIME | 3 | 0 | 0 | 0 | 0 |
| STATE | READY | READY | READY | READY | READY |

1. 为了清楚的观察各进程的调度过程，程序应将每个时间片内的情况显示出来，参照的具体格式如下：

RUNNING PROG：i

READY-QUEUE：-〉id1-〉id2

BLOCK-QUEUE：-〉id3-〉id4

= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = == = =

ID 0 1 2 3 4

PRIORITY P0 P1 P2 P3 P4

CUPTIME C0 C1 C2 C3 C4

ALLTIME A0 A1 A2 A3 A4

STARTBLOCK T0 T1 T2 T3 T4

BLOCKTIME B0 B1 B2 B3 B4

STATE S0 S1 S2 S3 S4

3、思考

（1）实际进程调度中，除了按调度算法选择下一个执行的进程外，还需要处理哪些工作？

（2）为什么对进程的优先数可按上述原则进行修改？

## 实验六 使用动态分区分配方式的模拟

1、实验目的

了解动态分区分配方式中使用的数据结构和分配算法，并进一步加深对动态分区存储管理方式及其实现过程的理解。

2、实验内容

（1）用C语言分别实现采用首次适应算法和最佳适应算法的动态分区分配过程alloc( )和回收过程free( )。其中，空闲分区通过空闲分区链来管理：在进行内存分配时，系统优先使用空闲区低端的空间。

（2）假设初始状态下，可用的内存空间为640KB，并有下列的请求序列：

•作业1申请130KB。

•作业2申请60KB。

•作业3申请100KB。

•作业2释放60KB。

•作业4申请200KB。

•作业3释放100KB。

•作业1释放130KB。

•作业5申请140KB。

•作业6申请60KB。

•作业7申请50KB。

•作业6释放60KB。

请分别采用首次适应算法和最佳适应算法，对内存块进行分配和回收，要求每次分配和回收后显示出空闲分区链的情况。

3、思考

（1）采用首次适应算法和最佳适应算法，对内存的分配和回收速度有什么不同的影响？

（2）如何解决因碎片而造成的内存分配速度降低的问题？

## 实验七 请求调页存储管理方式的模拟

1、实验目的

通过对页面、页表、地址转换和页面置换过程的模拟，加深对请求调页系统的原理和实现过程的理解。

2、实验内容

（1）假设每个页面中可存放10条指令，分配给一作业的内存块数为4。

（2）用C语言模拟一作业的执行过程。该作业共有320条指令，即它的地址空间为32页，目前它的所有页都还未调入内存。在模拟过程中，如果所访问的指令已经在内存中，则显示其物理地址，并转下一条指令。如果所访问的指令还未装入内存，则发生缺页，此时需记录缺页的次数，并将相应页调入内存。如果4个内存块中均已装入该作业，则需进行页面置换。最后显示其物理地址，并转下一条指令。在所有320条指令执行完毕后，请计算并显示作业运行过程中发生的缺页率。

（3）置换算法：请分别考虑OPT、FIFO和LRU算法。

（4）作业中指令的访问次序按下述原则生成：

•50%的指令是顺序执行的。

•25%的指令是均匀分布在前地址部分。

•25%的指令时均匀分布在后地址部分。

具体的实施办法是：

1. 在[0，319]之间随机选取一条起始执行指令，其序号为m；
2. 顺序执行下一条指令，即序号为m+1的指令；
3. 通过随机数，跳转到前地址部分[0，m-1]中的某条指令处，其序号为m1；
4. 顺序执行下一条指令，即序号为m1+1的指令；
5. 通过随机数，跳转到后地址部分[m1+2，319]中的某条指令处，其序号为m2；
6. 顺序执行下一条指令，即序号为m2+1的指令；
7. 重复跳转到前地址部分、顺序执行、跳转到后地址部分、顺序执行的过程，直至执行320条指令。

## 实验八 简单文件系统的实现

1、实验目的

通过具体的文件存储空间的管理、文件的物理结构、目录结构和文件操作的实现，加深对文件系统内部功能和实现过程的理解。

2、实验内容

（1）在内存中开辟一个虚拟磁盘空间作为文件存储器，在上面实现一个简单单用户文件系统。退出时应该将该虚拟文件系统保存到磁盘上，以便下次可以再将它恢复到内存对虚拟磁盘空间中。

（2）文件存储空间对分配可以采用显式链接分配或者其他的办法。

（3）空闲空间的管理可以选择位示图或者其他的办法。如果采用位示图来管理文件存储空间，并采用显式链接分配方式，那么可以将位示图合并到FAT中。

（4）文件目录结构采用多级目录结构。为了简单起见，可以不使用索引结点，其中的每个目录项应包含文件名、物理地址、长度等信息，还可以通过目录项实现对文件对读和写的保护。

（5）要求提供以下有关的操作：

√ format：对文件存储器进行格式化，即按照文件系统对结构对虚拟磁盘空间进行布局，并在其上创建根目录以及用于管理文件存储空间等的数据结构。

√mkdir：用于创建子目录；

√rmdir：用于删除目录；

√ls：用于显示目录；

√cd：用于更改当前目录；

√create：用于创建文件；

√open：用于打开文件；

√close：用于关闭文件；

√write：用于写文件；

√read：用于读文件

√rm：用于删除文件。