考察知识点:逻辑地址和物理地址的转换,访问时间

6、(8分) 某请求分页系统,用户空间为32KB,每个页面1KB,主存16KB。某用户程序有7页长, 某时刻该用户进程的页表如下:

页号	物理块号	是否在 TLB
0	8	是
1	7	是
2	4	否
3	10	否
4	5	否
5	3	是
6	2	是

- (1) 计算两个逻辑地址: 0AC5H、1AC5H 对应的物理地址,给出计算过程。
- (2) 已知主存的一次存取为 10ns,对于 TLB 表(快表)的查询时间可以忽略,假设地址转换时,先访问 TLB,若 TLB 未命中时再访问页表(忽略 TLB 更新时间),则访问上述两个逻辑地址共耗费多少时间?

### 参考解答:

(1)

逻辑地址和物理地址之间的转换规则:

偏移量不变, 页号转换成物理块块号

每个页面1KB,即一个页面的大小为1KB(2^10B),

∴偏移量占10位

用户空间占32KB(2^15)-->逻辑地址占15位

∴页号位数=逻辑地址位数-偏移量位数=15-10=5

主存16KB(2^14)-->物理地址占14位,

∴物理块块号位数=物理地址位数-偏移量位数=14-10=4

0AC5H转换成二进制为:

000 1010 1100 0101

页号为000 10, 转换成十进制数为2,

查页表得: 页号为2, 对应的物理块块号为4,

将4转换成4位的二进制数为0100

又: 偏移量为10 1100 0101

∴0AC5H对应的物理地址为01 0010 1100 0101,

转换成十六进制数为(12C5)H

1AC5H转换成二进制为:

001 1010 1100 0101

页号为001 10, 转换成十进制数为6,

查页表得: 页号为6, 对应的物理块块号为2,

将2转换成4位的二进制数为0010

又: 偏移量为10 1100 0101

∴1AC5H对应的物理地址为00 1010 1100 0101,

转换成十六进制数为(0AC5)H

(2)

访问0AC5H,该页表项不在TLB中,因此需访问一次内存的页表取出物理块块号,跟偏移量拼接成物理地址后,还需访问一次内存,取操作数。故耗费的时间为20(10+10)ns。

访问1AC5H,该页表项在TLB中,因此将TLB中查询出的物理块块号和偏移量拼接成物理地址后,需访问一次内存,取操作数。故耗费的时间为10ns。

考察知识点:磁盘调度算法

- 3. (10 分) 在一个系统中,某时刻有如下柱面请求序列: 15,20,9,16,24,13,29。此时磁头位于 15 柱面,移臂方向正在向着磁道号增加的方向移动,请回答以下问题:
  - (1) 若移臂调度采用最短寻道时间优先算法,请给出调度顺序并计算寻道距离。
  - (2) 若移臂调度采用 SCAN 算法,请给出调度顺序并计算寻道距离。
  - (3) 分析最短寻道时间优先算法为什么在现代操作系统中使用较少的原因。
- (4) 针对 SCAN 算法,如果希望进一步缩短平均寻道距离,降低寻道时间,请给出一种合理的优化思路。

# 参考解答:

(1)

最短寻道时间优先算法(SSTF、Shortest Seek Time First):

SSTF调度算法会选择这样的进程,其要求访问的磁道与当前磁头所在的磁道距离最近,

以使每次的寻道时间最短,但这种算法不能保证平均寻道时间最短。

此时磁头位于15柱面,则调度顺序为

16、13、9、20、24、29

寻道距离=(16-15)+(16-13)+(13-9)+(20-9)+(24-20)+(29-24)

=1+3+4+11+4+5=28

(2)

SCAN算法类似于电梯的运行规律

不仅会考虑欲访问的磁道与当前磁道间的距离,还会优先考虑磁头当前的移动方向。

- :: 移臂方向正在向着磁道号增加的方向移动
- :.调度顺序为

16, 20, 24, 29, 13, 9

寻道距离=(16-15)+(20-16)+(24-20)+(29-24)+(29-13)+(13-9)

=1+4+4+5+16+4=34

(3)

原因:

最短寻道时间优先算法(SSTF)的实质是基于优先级进行调度,

可能会导致优先级低的进程出现"饥饿"现象。

# (4)(共2分)说出一种合理的方法即可,比如可以考虑将相邻的磁盘请求进行合并等。

考察知识点:索引文件

- 5. (10 分) 某文件系统采用二级索引文件结构,假定文件索引表用 5B 存放一个磁盘块的块号,磁盘块的大小为 1KB。请问:
- (1) 该文件系统能支持的最大文件大小是多少字节? 能管理的最大磁盘空间是多大?
- (2)一个文件系统,它能支持的文件大小与哪些因素有关?能管理的最大磁盘空间大小 又与哪些因素有关?

参考解答:

(1)

由题意得:

一个文件索引表能存放的磁盘块块号的个数=1KB/5B

=1024B/5B=204...4

即最多能存放204个磁盘块块号

因该文件系统采用二级索引文件结构,

(主索引指向第二级索引,第二级索引指向磁盘块),

一个磁盘块大小为1KB

::能支持的最大文件大小为

204\*204\*1KB=41616KB

最大磁盘空间与磁盘块号个数和磁盘块大小相关:

磁盘块块号占5B,即占5\*8=40位

∴最多有2^40个磁盘块号,即最多有2^40个磁盘块

又:一个磁盘块的大小为1KB

∴能管理的最大磁盘空间是2^40\*1KB=2^40KB=2^50B

∵1PB=1024TB

1TB=1024GB

1GB=1024MB

1MB=1024KB

1KB=1024B

∴1PB=2^50B

:能管理的最大磁盘空间是1PB

(2)

(2)一个文件系统能支持的最大文件与文件的物理结构、盘块大小等有关。对于索引文件系统而言,它能支持的文件大小与磁盘块的大小、磁盘块号所占的字节数、间接索引层次数等有关。(3分)

能管理的磁盘空间大小,与磁盘块的大小和磁盘块号所占的字节数有关。(2分)

考察知识点: 进程同步

3. (11分)系统中有3个进程A,B,C,相互协作完成对一组数据的分类汇总工作。进程A循环地从一个存着整数的文件中不断地读出数据,若该整数为偶数则放入1号缓冲区;若该整数为奇数则放入2号缓冲区。进程B不断的从1号缓冲区读出数据并累计求和;进程C不断的从2号缓冲区读出数据并累计求和。每个缓冲区最多都可以放5个整数。试用信号量的PV操作实现三个进程间的同步关系。

```
参考解答:
定义如下信号量:
empty1=5,表示1号缓冲区还能存放的整数个数,一开始1号缓冲区为空,
::1号缓冲区能存放的整数个数为5
empty2=5,表示2号缓冲区还能存放的整数个数,一开始2号缓冲区为空,
:.2号缓冲区能存放的整数个数为5
full1=0,表示1号缓冲区已经存放的整数个数,一开始1号缓冲区为空,
::1号缓冲区已经存放的整数个数为0
full2=0,表示2号缓冲区已经存放的整数个数,一开始2号缓冲区为空,
:.2号缓冲区已经存放的整数个数为0
mutex1=1,表示锁信号量,用来使得进程A和进程B互斥地访问1号缓冲区
mutex2=1,表示锁信号量,用来使得进程A和进程C互斥地访问2号缓冲区
A(){
   while(1){
      读出数据
      if(数据%2==0){
         P(empty1);
         P(mutex1):
         将数据放入1号缓冲区
         V(mutex1);
         V(full1);
      }else{
         P(empty2);
         P(mutex2);
         将数据放入2号缓冲区
         V(mutex2);
         V(full2);
      }
   }
}
B(){
   while(1){
      P(full1);
      P(mutex1);
      从1号缓冲区读出数据,并累计求和
      V(mutex1);
      V(empty1);
   }
}
C(){
   while(1){
      P(full2);
      P(mutex2);
      从2号缓冲区读出数据,并累计求和
      V(mutex2);
```

```
V(empty2);
}

void main(){
  cobegin
    A();
    B();
    C();
  coend
}
```

考察知识点: 分页存储管理

- 2. (12 分)在一个页式存储管理系统中,有 J1、J2 和 J3 共 3 个作业同时驻留内存,其中 J3 有 4 个页面,被分别装入到内存的 3、4、6、8 块中。假定页面大小为 1KB,内存容量为 32KB。请完成以下问题:
- (1) 写出 J3 的页表的内容。
- (2) 当 J3 在 CPU 上运行时, 执行到其地址空间的 500 号单元处的一条指令: MOV 1500, 3100

请问执行该指令时需要访问哪些逻辑地址?对应的物理地址分别是多少?给出计算过程。

(3) 若一次内存的访问时间为 100ns, 快表命中率为 75%, 快表访问时间忽略不计,则完成 J3 上述指令的访存要求共需要多少时间?

```
参考解答:
(1)
```

页表:每一个页表项记录了页号对应的物理块块号

J3的页表内容如下图所示:

【分页存储管理中,页号从0开始编号】

页号	物理块块号
0	3
1	4
2	6
3	8

(2) 要访问的逻辑地址有500、1500、3100

#### MOV指令的格式是什么? 🕤

MOV 指令将源操作数复制到目的操作数。作为数据传送(data transfer)指令,它几乎用在所有程序中。在它的基本格式中,第一个操作数是目的操作数,第二个操作数是源操作数:

MOV destination, source

页号=逻辑地址/页面大小并向下取整偏移量=逻辑地址 MOD 页面大小页面大小占1KB(1024B)

# 逻辑地址500:

页号=500/1024=0

偏移量=500 MOD 1024=500

查找页表,得页号为0,对应的物理块块号为3,

::物理地址=物理块号\*物理块大小+偏移量

又: 物理块的大小=页的大小【书本P154,分页存储管理方式,页和块的大小相同】

∴物理地址=3\*1024+500=3572

#### 逻辑地址1500:

页号=1500/1024=1

偏移量=1500 MOD 1024=476

查找页表,得页号为1,对应的物理块块号为4,

::物理地址=4\*1024+476=4572

### 逻辑地址3100:

页号=3100/1024=3

偏移量=3100 MOD 1024=28

查找页表,得页号为3,对应的物理块块号为8,

∴物理地址=8\*1024+28=8220

#### (3)

由于没有缺页,对每个地址的内存访问时间相同,

访问逻辑地址500:

::引入快表的分页存储管理方式中,有效访问时间由两部分组成:

②形成物理地址的时间【查快表命中的时间+查快表未命中,访问页表获取物理块块号的时间】 ②取操作数的时间

即,引入快表后的内存有效访问时间= $\alpha \times \lambda + (1-\alpha) \times (t+\lambda) + t$ 

【其中 $\alpha$ 表示快表的命中率, $\lambda$ 表示查快表的时间,t表示访问一次内存的时间】

据题意,  $\alpha$ =0.75,  $\lambda$ =0, t=100ns, 代入上式, 得:

访问时间=(1-0.75) X 100 + 100=125ns

完成33上述指令,需要进行3次地址的访问,故

3个逻辑地址的访问时间: 125\*3=375ns

没必要继续刷题了,平时老师布置了这么多习题,全消化掌握了即可。剩余的时间,我准备看平时练习。

最后预祝各位操作系统能取得令自己满意的分数。 🐉 🎉