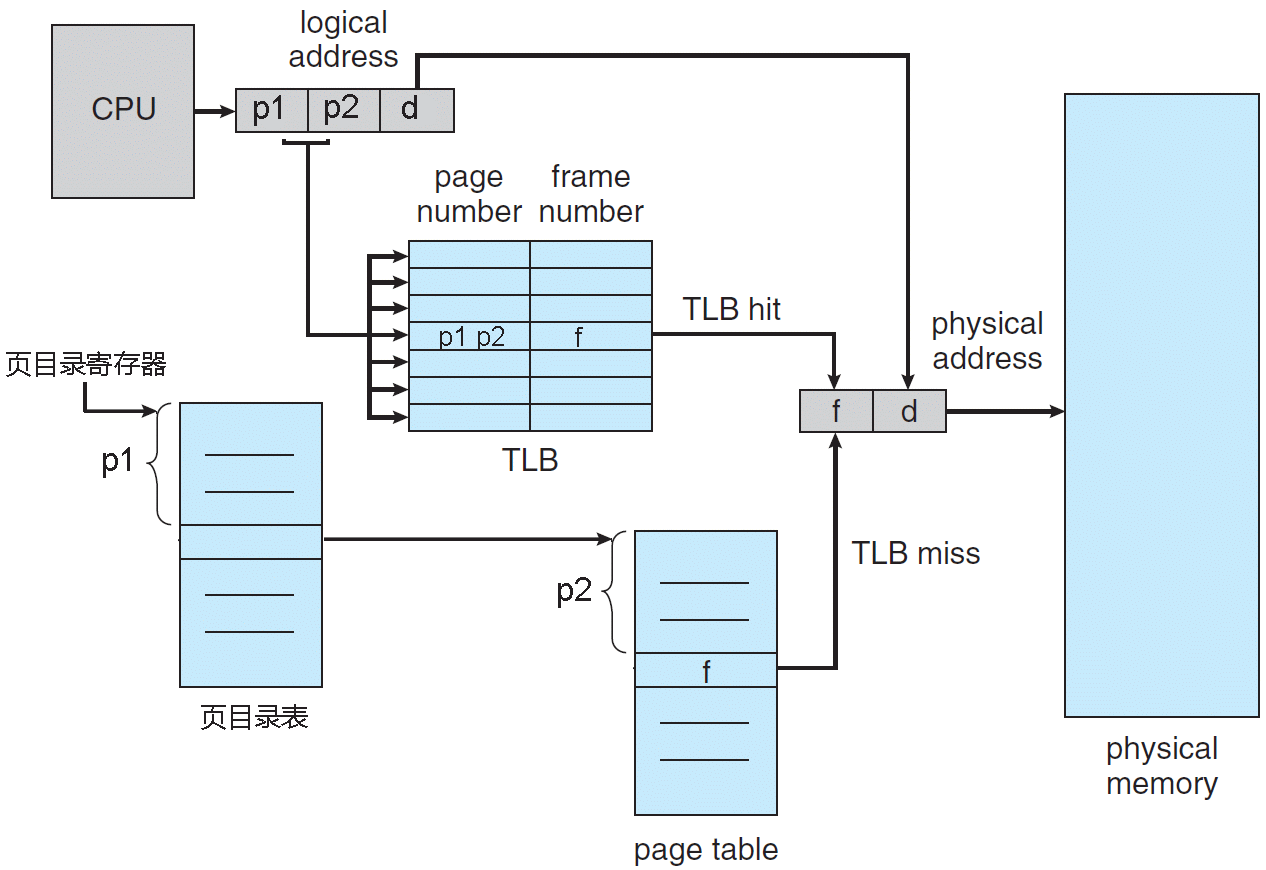
OS 第8-9次作业 部分习题解读

2021.6.10

**第8题：**

Assume a system uses 2-level paging and has a TLB hit ratio of 90%. It requires 15 nanoseconds to access the TLB, and 85 nanoseconds to access main memory. The effective memory access time in nanoseconds is [填空1]

注：TLB命中的是完整的逻辑页号(含一二级页号两部分)到物理帧号的最终转换结果(无需二次分别命中二级页表才得到结果)。也没有TLB仅命中一级页表或仅命中二级页表的情况



**B4**

**B3**

**B2**

**A2**

**A1/B1**

2级页表 + TLB，分TLB命中与未命中两种情况：

A1：TLB命中，得到帧号f；

A2：帧号f与页内偏移量拼接后，得到物理地址，按物理地址访问主存

B1：TLB未命中；

B2：用p1作为索引，访问页目录表(页目录表起始地址在相应的硬件寄存器内)，得到页表起始地址；

B3：用p2作为索引，访问页表，得到帧号f；

B4：帧号f与页内偏移量拼接后，得到物理地址，按物理地址访问主存

TLB命中率假设为p，则：

**有效内存访问时间 = p \* (TLB时间 + 物理内存访问时间)**

**+ (1 – p) \* (TLB时间 + 页目录表访问时间 + 页表访问时间 +物理内存访问时间)**

页目录表访问时间，页表访问时间，与物理内存访问时间都是一样的85ns，因为不管是页目录表，还是页表，都是存储在主物理内存中的。

**有效内存访问时间 = 0.9 \* (15 + 85) + 0.1 \* (15 + 85 + 85 + 85) = 117 ns**

**第11题：**

一个进程的页表见下面的表格，页的大小为1024字节。为执行指令 MOV AX, [2560]和MOV BX, [4098]，计算逻辑地址2560和4098对应的物理地址。

页号 帧号

0 20

1 30

2 18

3 80

注：本题中所有数字均为10进制。[2560]表示访问逻辑地址为2560的存储器，AX及BX均为寄存器名，MOV为移动数据的汇编指令

MOV AX, [2560] 是X86的汇编指令，意思是，把逻辑地址为2560的内存数据读出来，并传给CPU内部的寄存器AX(16位，但本题目中数据未跨页到达下一个逻辑页号，因此与字长无关)。另一条指令类似，只不过寄存器为BX

MOV AX, [2560] 与下面的c语句含义类似：

short AX；// 假设在这里的c中，short为16位长

AX = \*((short \*)(2560));

1.

2560 / 1024 = 2，页号2对应帧号18；

2560 % 1024 = 512

物理地址：18 \* 1024 + 512 = 18944

2.

4098 / 1024 = 4 > 3，页表中没有4号页，因此地址溢出

思考：把MOV AX, [2560] 改为 MOV AX, [4095]，会发生什么？

**第12题：**

某程序访问下列页面:

0, 6, 1, 7, 8, 1, 8, 6, 7, 8, 7, 2, 3, 4, 3, 5, 7, 6

如果程序有4个页框可用，且使用下列算法，将会产生多少次缺页？并给出替换过程（一开始4个页框均为空，缺页次数包括填充这些空页框的缺页次数）？

(1) LRU 替换算法

(2) 最佳替换算法

LRU: 11次

0 6 1 7 8 1 8 6 7 8 7 2 3 4 3 5 7 6

0 0 0 0 8 8 8 4 4 4 6

6 6 6 6 6 3 3 3 3 3

1 1 1 2 2 2 2 7 7

7 7 7 7 7 5 5 5

OPT: 9次

0 6 1 7 8 1 8 6 7 8 7 2 3 4 3 5 7 6

0 0 0 0 8 2 3 3 5

6 6 6 6 6 6 6 6

1 1 1 1 1 4 4

7 7 7 7 7 7

注：正式答卷时水平方向请间隔空白，最好如教材中那样页框是带实线的框

注意OPT置换算法中，牺牲页为未来最晚使用的页。若一个页在引用串中不再出现，那就等效为距离无穷远。若页框中同时存在不止一个无穷远的页，具体置换哪个页，算法没有规定，我们可以简单地选第一个无穷远的页即可。

**第13题：**

一进程指令片段如下表（以下地址均为16进制。访存地址包括指令地址，以逗号分隔的为多地址）：

指令序号 访存地址

1 F0, E0

2 F4,ED,EA

3 F8

4 FB,EE,ED

5 FE,EB,BA

当前PC指向第一条指令，该进程所有页面都不在内存，分配了**2**帧，帧号是**A**和**B**(16进制)。帧的大小是16字节。采用局部置换，先进先出(FIFO)的置换算法。给出该进程访问**物理**内存的**物理地址**序列

帧的大小是16字节，那么以第3条指令为例，访存地址F8中，最后4个比特为页内偏移量，高4位为页号。也就是F8是第F页的，页内偏移量为8

1 A0, （1分） B0, （1分）

2 A4, （1分） BD, （1分） BA, （1分）

3 A8, （1分）

4 AB, （1分） BE, （1分） BD, （1分）

5 执行第五条指令时，因只有两帧，指令需要三页，将陷入无穷的换页。（1分）

注意执行第2及第4条指令时，虽然也有3个访存地址，但都有2个地址是在同一页内(ED,EA是同一页，EE,ED也是同一页)，因此不会出现帧数不够用的现象