

4.1

① 有程序运行时才需要的数据, 也有一直需要的数据, 易失性不同使得需要分级

② CPU 访问速度远大于内存响应速度, 故需要层级来减小差距

③ 不同类型存储器的成本不同, 需综合考虑

4.2

1. 过大的页面起意为程序所需, 会造成较多额外的占用

2. 当页面较大时其与内存的交换也较慢

3. 若页过小, 会造成一些很小的页不能被任何程序使用

4. 过小的页 页数较多, 不利于内存管理

4.3

1. ① V 标志该页表是否有效

② RWX 代表该页表的权限为读, 写, 执行, 1 有效

③ V 代表在用户模式下其是否可用

④ G 代表全局映射

⑤ A 代表该页是否被访问过, D 代表其是否被写过。

2. 若更改标志位, 会导致页表混乱, 从而使得操作系统崩溃

3. 为一个指向下一级页表的指针

4.4 (1) 其与上述作用类似, 但其能更为精确地进行保护

(2) A 为 0 不允许映射, 其还指定了三种映射模式

A	Name	Description
0	OFF	Null region (disabled)
1	TOR	Top of range
2	NA4	Naturally aligned four-byte region
3	NAPOT	Naturally aligned power-of-two region, ≥ 8 bytes

1 为锁定标志位, 其对应锁定的内存的访问无效

4.5 (1) 如果全映射, 有

$$2^{14} B / 4 \times 1024 B = 2^{32} \text{ (页表)}$$

$$2^{32} \times 2^3 = 2^{35} B$$

$$\text{则需要 } 2^{35} B = 2^{40} \text{ KiB} = 2^{30} \text{ MiB} = 2^{25} \text{ GiB} = 2^{15} \text{ TiB}$$

的空间有储

(2) 若使用了 48 位, 有

$$2^{45-11} = 2^{34} = 2^9 \text{ GiB}$$

(3) ① 其划分多级, 使得每一级页表都较小, 内存消耗较少

② 不需要时不创建页表, 从而降低开销

③ 期间可以实现进程共享与硬连接类似, 能降低实际存储空间