存储器层次结构练习题

1. 已知一个硬盘容量为1TB，由两个盘（4个盘面）构成，每个盘面由3200个磁道构成，每个磁道平均有128个扇区。由于磁记录密度的提升，新工艺维持相同的容量下，只需要一个盘（2个盘面），每个盘面的磁道数为3200，则新工艺要求每个磁道多少个扇区？

新工艺每个磁道的扇区数==256

1. 请写出磁盘扇区访问时间的计算公式，并简单解释寻道时间、旋转时间、传送时间。

磁盘扇区访问时间=寻道时间+旋转时间+数据传输时间。

寻道时间：磁头由一个柱面移动到另一个柱面的时间。

旋转时间：经过磁盘旋转，目标扇区到达磁头下的时间。

数据传输时间：传输每个扇区所需要的时间。

1. 如果cache块（行）的大小为64字节，cache总容量为1KB，请问cache的块数有多少？

cache的块数

1. 请简述直接映射、组相连映射、全相联映射的原理

直接相联(Direct)：每个主存块映射到Cache的固定行。

直接映射是一种简单的映射方法，将主存地址空间分成多个块，并将缓存分为相同数量的行。每个主存块只能映射到缓存中的特定行。具体的映射规则是，主存地址的一部分用于标识缓存中的行号，另一部分用于标识行内的偏移量。当需要访问某个主存地址时，先计算出对应的行号，然后检查该行是否已经存储了待访问的数据。如果已经存在，则命中缓存，直接返回数据；如果不存在，则未命中缓存，需要从主存中加载数据并替换掉缓存中的某一行。

组相联(Set Associate)：每个主存块映射到Cache固定组中任一行。

组相连映射是在直接映射的基础上进行改进的一种映射方法。相比直接映射，组相连映射在缓存中引入了多个行的集合，称为组。每个主存块可以映射到多个组中的某一行。具体的映射规则是，主存地址的一部分用于标识组号，另一部分用于标识行内的偏移量。当需要访问某个主存地址时，先计算出对应的组号，然后在该组中查找是否已经存储了待访问的数据。如果已经存在，则命中缓存，直接返回数据；如果不存在，则未命中缓存，需要从主存中加载数据并替换掉该组中的某一行。

全相联(Full Associate)：每个主存块映射到Cache的任一行。

全相联映射是最灵活的映射方法之一。在全相联映射中，主存地址可以映射到任意的缓存行。具体的映射规则是，主存地址的所有位都用于标识缓存中的行号。当需要访问某个主存地址时，需要遍历整个缓存以查找是否已经存储了待访问的数据。如果已经存在，则命中缓存，直接返回数据；如果不存在，则未命中缓存，需要从主存中加载数据并替换掉缓存中的某一行。

1. 请简单分析一下各项因素单独变化时对cache命中率的影响：cache容量大小变化、cache块大小、组的大小。

Cache容量大小：容量提高会提升命中率，增加命中时间。

Cache块大小：块大小提高有利于提高命中率，但随着块数量减少，命中率也会下降。

组大小：较大的组降低了高速缓存由于冲突不命中出现抖动的可能性。

1. 什么是数据访问的时间局部性和空间局部性？

时间局部性：最近被访问的数据或指令在未来可能还会被访问。

空间局部性：最近访问地址附近的区域在不久可能还会被访问。

链接练习题

1. 将C语言程序到生成可执行文件需要经过哪些步骤？各自的输出是什么？

预处理：主要对源代码进行一些宏替换和头文件包含等操作。main.c → main.i。

编译：将预处理后的代码翻译成汇编语言代码。main.i → main.s。

汇编：将汇编语言代码转化为机器码指令。main.s → main.o。

链接：将目标文件及所需的库函数等链接成最终的可执行文件。main.o → main.out。

1. 已知如下程序

#include <stdio.h>

int var1=1;

int var2;

int myfun1()

{ return var2+var1;

}

int main(){

{ int c=4;

var1=var2+c+myfun1()+myfun2();

printf(“result var1=: %d \n”,var1);

return;

}

请问执行gcc main.c myfun2.c命令时，在链接开始时扫描完main.o之后，未定义符号集U,已定义符号集D和代码集合E各自包含什么？

1. 已知main.o中定义了main()、调用了myfuns.o里的m1()和m2()函数，且调用了printf()函数。在libc.a里的printf.o定义了printf()，libc.a里的sqrt.o定义了sqrt()。在自己编写的myfuns.o里定义了m1()和m2()，其中m1()又调用了sqrt()，m2()则调用了printf()。请问编译时，gcc main.o libc.a myfuns.o和gcc main.o myfuns.o libc.a能否正确产生可执行文件？
2. 请问下面main.c和fun1.c两个源文件中出现多处同名变量，是否能正确进行链接并产生可执行文件？如果能，请给出下面代码运行结果；如果不能，则说明理由。

//main.c

int a=10;

int main()

{ int b=100;

a=a+fun1(b);

printf (“a=: %d\n”,a);

return ;

}

//fun1.c

int b=81;

int fun1(int b)

{

int a=60;

a=a+sqrt(b);

return a;

}

期末复习题

1. 已知一个硬盘容量为1TB，由两个盘（4个盘面）构成，每个盘面由3200个磁道构成，每个磁道平均有128个扇区。由于磁记录密度的提升，新工艺维持相同的容量下，只需要一个盘（2个盘面），每个盘面的磁道数为3200，则新工艺要求每个磁道多少个扇区？
2. 请写出磁盘扇区访问时间的计算公式，并简单解释寻道时间、旋转时间、传送时间。
3. 什么是数据访问的时间局部性和空间局部性？
4. 请简单说明什么是连接器相关的强符号、弱符号？

强符号：函数和已经初始化的全局变量。

若符号：未初始化的全局变量。

1. 对于以下代码：
2. int a=100;
3. short proc(int inputarg)
4. { int a;

a=inputarg;

return a+0;

1. }

请说明边第1行变量a和3行定义变量a的作用域。如果将第3行变量类型修改为static int，则第一次调用函数proc时传入参数inputarg的值为12，此时返回值为多少？再次调用时传入inputarg为5，则第二次调用的返回值是多少？

1. 对于以下myfile.c程序代码

int hello1;

char myfun(int a)

{ int b=10;

hello1=b+a;

hello1=hello1\*f1(a);

printf(“%d\n”,hello1);

return hello1;

}

请指出那些符号是链接器符号，哪些符号需要重定位？