

1. 以下哪个工作**不属于**动态链接器完成的__ B __?

- A、完成对引用的动态库函数在进程空间的布局。
- B、完成动态库函数的编译和汇编。
- C、完成可执行文件对动态库函数引用的符号解释。
- D、完成可执行文件对动态库函数引用的地址确定。

2. ELF 格式可用来保存哪些类型的目标文件?

可重定位目标二进制文件；可执行文件，共享库

3. 对以下两个源代码进行符号解析时，以下描述**错误**的是__ B __?

```
/*m1.c*/
int x = 100;
int p1(void);
int main()
{   x = p1();
    return x;
}
```

```
/*m2.c*/
float x;
static main = 1;
Int p1()
{   int p1 = main + (int) x;
    return p1;
}
```

- A、m1 中对 x 的两处引用都与模块 m1 中定义的 x 相绑定；
 - B、m2 中对 x 的一处引用与 m2 中定义的 x 相绑定；
 - C、m2 中的变量 p1 和函数 p1 被分配在不同的存储空间中；
 - D、虽然 x、main、和 p1 都出现了多次定义，但是不会发生链接错误。
4. 假设调用关系如下：hello.c 调用 libx.a 和 liby.a 中的函数，liby.a 调用了 libz.a 的函数，libz.a 调用了 libx.a 和 liby.a 中的函数，以下编译正确的是 C 。

- A、gcc -o hello hello.c ./libx.a ./liby.a ./libz.a ./libx.a
- B、gcc -o hello ./libx.a ./liby.a ./libz.a ./libx.a ./liby.a hello.c
- C、gcc -o hello hello.c ./libx.a ./liby.a ./libz.a ./libx.a ./liby.a
- D、gcc -o hello hello.c ./libx.a ./liby.a ./libz.a

5. 下述说法中，**错误**的是__ D __?

- A、SRAM 的相对访问时间比 DRAM 要短。
- B、DRAM 只有不断刷新的条件下信息才是不易失的。
- C、磁盘以扇区为大小来读写数据。
- D、SSD 固态硬盘以扇区为大小来读写数据。

6. 某企业计划采购一款磁盘，假设该款磁盘的参数如下表所示，请计算该款磁盘上一个扇区的平均访问时间（以 ms 为单位）。

性能参数	数值
旋转速率	7200RPM
平均寻道时间	8ms
每条磁道的平均扇区数	600
缓存容量	128MB
接口类型	SATA 6Gb/s
单个盘片容量	1TB

平均旋转时间= $0.5 \times 1/7200 \times 60 \times 1000$ ms

数据传输时间= $1/7200 \times 60 \times 1000/600$ ms

一个扇区的平均访问时间=平均寻道时间+平均旋转时间+数据传输时间

7. 请分析以下代码是否局部性良好，并给出你的优化设计。请写出详细的分析过程。

```
#define N 10000
typedef struct{
    int LearningInvest[5];
    int FunInvest[5];
}PersonallInvest;
PersonallInvest undergraduates[N];

void UpdateInvestment (PersonallInvest *p, int n)
{
    int i, j;
    for (j = 0; j < 5; j++){
        for (i = 0; i < n; i++)
            p[i].LearningInvest[j] +=1000;
        for (i=0; i<n; i++)
            p[i].FunInvest[j] +=500;
    }
}
```

不具有良好的就不性。

```

#define N 10000
typedef struct{
    int LearningInvest[5];
    int FunInvest[5];
}PersonallInvest;
PersonallInvest undergraduates[N];

void UpdateInvestment (PersonallInvest *p, int n)
{
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < 5; j++)
            p[i].LearningInvest[j] +=1000;
        for (i=0; i<n; i++)
            p[i].FunInvest[j] +=500;
    }
}

```

8. 可重定位目标文件中机器代码、只读数据、未初始化的全局变量以及已初始化数据通常分别保存在哪些节里？

Text节，rodata节，bss节,data节

9. 已知一个 C 语言程序有两个源文件 `main.c` 和 `test.c`，其内容如下：

```
/*main.c*/
int sum();
int a[4] = [3, 2, 33, 2];
extern int val;
int main( )
{
    val = sum();
    return val;
}
```

```
/*test.c*/
extern int a[ ];
int val = 0;
int sum( )
{
    int i;
    for (i = 0; i < 4; i++)
        val *= a[i];
    retrun val;
}
```

对于编译后生成的可重定位目标文件 `test.o`，请填写以下表中各符号的情况，

说明每个符号是否出现在 `test.o` 的符号表（`.symtab`）中。如果是，定义该符号的模块是 `main.o` 还是 `test.o`？该符号的类型是全局、外部，还是本地符号？该符号出现在相应定义模块的那个节（`.text`、`.data` 或 `.bss`）？

符号	是否链接符号	在哪个模块定义	符号类型	所在节
a	是	Main.o	全局	Data
val	是	Test.o	全局	Bss
sum	是	Test.o	全局	Text
i	否		局部变量	
main	是	Main.o	全局	Text