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。

1. ELF格式可用来保存哪些类型的目标文件？

可重定位二进制目标文件(.o)，可执行文件，动态库(.so)

1. 对以下两个源代码进行符号解析时，以下描述**错误**的是 B ？

/\*m2.c\*/

float x;

static main = 1;

Int p1()

{ int p1 = main + (int) x;

return p1;

}

/\*m1.c\*/

int x = 100;

int p1(void);

int main()

{ x = p1();

return x;

}

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。

1. 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调动了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

1. 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。

1. 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 |

8ms+0.5\*1/7200\*60\*100ms+1/7200\*60\*100/600ms

1. 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];  }PersonalInvest;  PersonalInvest undergraduates[N];  void UpdateInvestment (PersonalInvest \*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。优化设计为：

#define N 10000

typedef struct{

int LearningInvest[5];

int FunInvest[5];

}PersonalInvest;

PersonalInvest undergraduates[N];

void UpdateInvestment (PersonalInvest \*p, int n)

{

int i, j;

for (i = 0; i < n; i++) {

for (j = 0; j < 5; j++)

p[i].LearningInvest[j] +=1000;

for (j = 0; j < 5; j++)

p[i].FunInvest[j] +=500;

}

}

1. 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 .rodata .bss .data

1. 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：

/\*test.c\*/

extern int a[ ];

int val = 0;

int sum( )

{

int i;

for (i = 0; i < 4; i++)

val \*= a[i];

retrun val;

}

/\*main.c\*/

int sum();

int a[4] = [3, 2, 33, 2];

extern int val;

int main( )

{

val = sum();

return 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 | test.o外部 | Main.o的.data |
| val | 是 | test.o | test.o全局 | test.o的bss |
| sum | 是 | test.o | 全局 | Txt |
| i | 否 | / | / | / |
| main | 是 | main.o | 全局 | Main.o的.txt |