Homework 13

Problem 1

TA wrote a simple program and there is a call to function ***printf*** which is from a shared library. And after using gdb, TA found that the start address of **_GLOBAL_OFFSET_TABLE_** is **0x804a000**. And the partial .PLT(Procedure Linkage Table) after linking is:

```
080482f0 <printf@plt>:
    80482f0:    ff 25 0c a0 04 08       jmp    *0x804a00c
    80482f6:    68 00 00 00 00          push   $0x0
    80482fb:    e9 e0 ff ff ff          jmp    80482e0 <_init+0x30>
```

1) What is the value stored in the address **0x804a00c before** first calling the printf() function? (NOTE: resolved as a 32-bit hexadecimal)

Answer: 0x80482f6

2) What is the index of printf() in _GLOBAL_OFFSET_TABLE_? (NTOE: The index starts from 0)

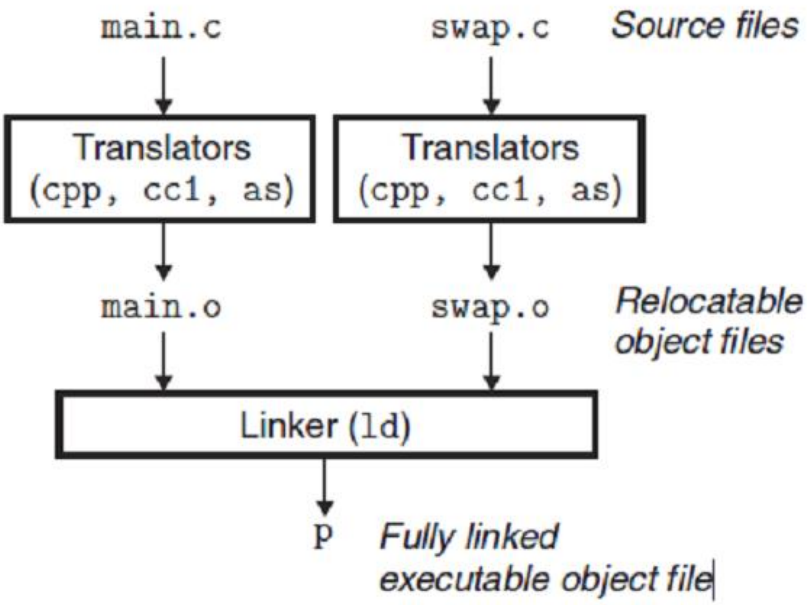
Answer: 3

Problem 2

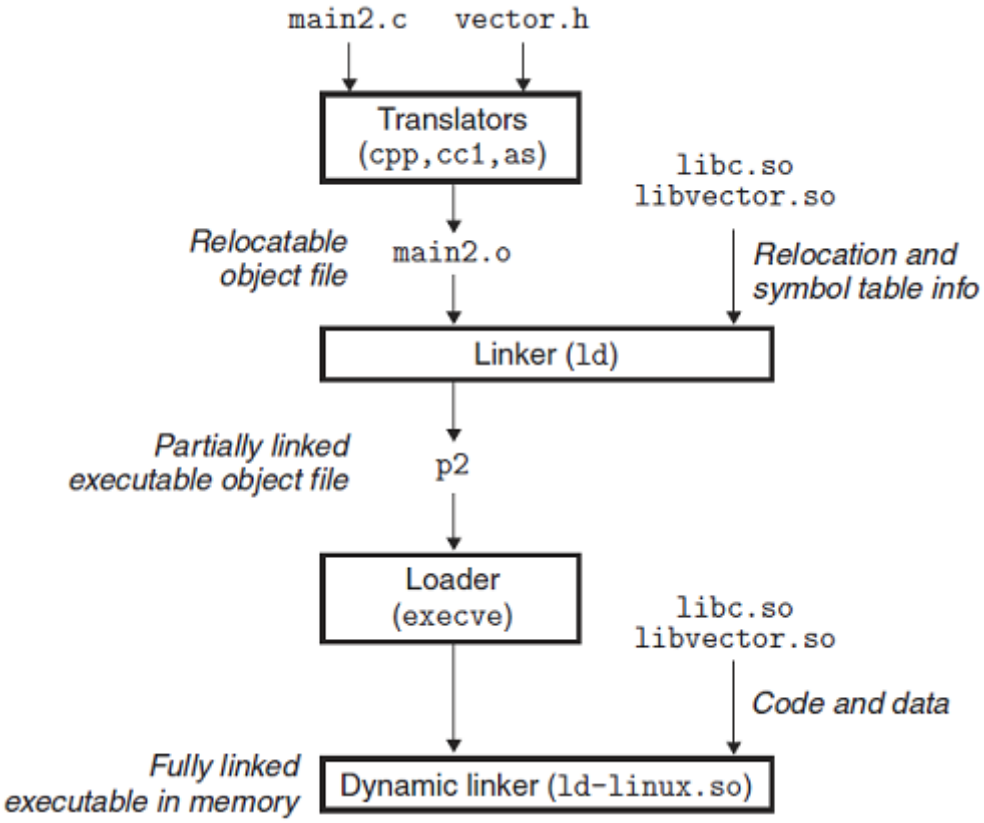
Express the difference between the procedure of static link and the procedure of dynamic link (with shared library). You can draw a picture to show that.

Answer:

Static link:



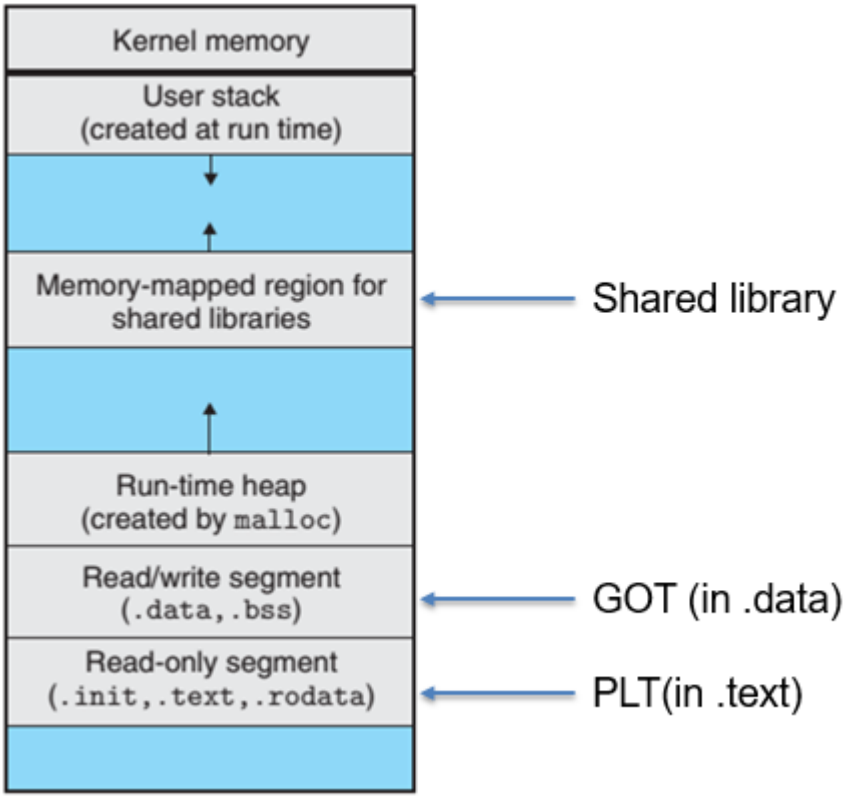
Dynamic link:



Problem 3

1. Draw a Linux run-time memory image to show the location of GOT, PLT and shared library.
2. Express the function of PLT and GOT. And how to use them?
3. When the GOT is generated, and when the items of it are relocated?(Two situations: PIC Data References and PIC Function Calls)

Answer:

1. 

2. GOT 作用是把位置无关的地址重定位到绝对地址；PLT 的作用是把位置独立的函数调用重定向到绝对位置。
   PIC 数据引用时，通过位置无关代码，使程序跳转到 GOT 表中的相应条目，通过 GOT 间接引用每个全局变量。
   PIC 函数调用时，将被调用函数绑定到 PLT 表中的相应条目中，并从条目的第一条指令开始执行，从而得到绝对地址。

3. GOT 表由静态链接器生成，在动态链接时由动态链接器重定位。