Pg391

3

1)

在标准输出设备上显示字符串`Hello, world.`.

2)

执行第16、20行的`int $0x80`指令会从用户态到内核态。

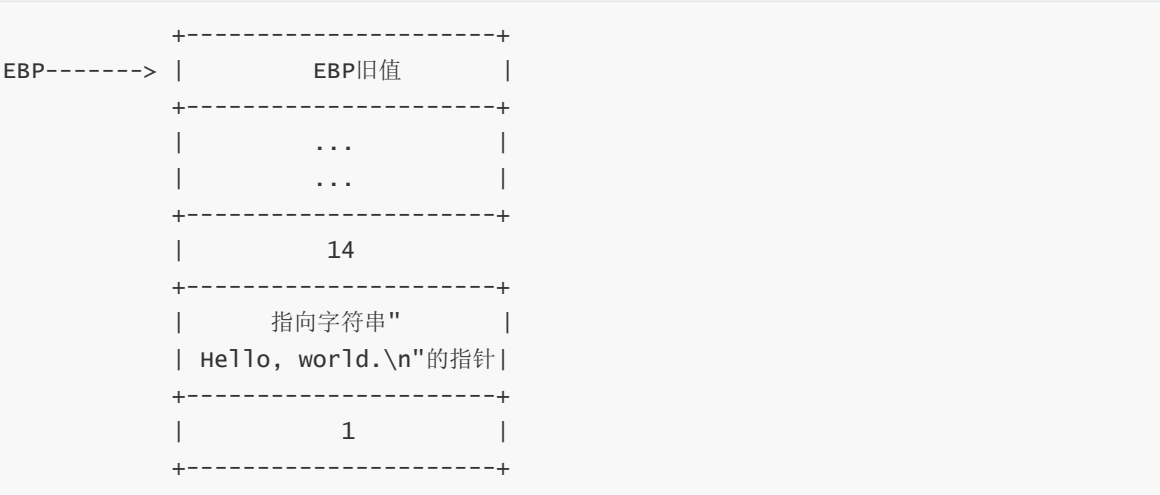
3)

第16行调用了0x4号系统调用`write`，服务例程为`sys_write()`函数；第20行调用了0x1号系统调用`exit`，服务例程为`sys_exit()`函数.

4

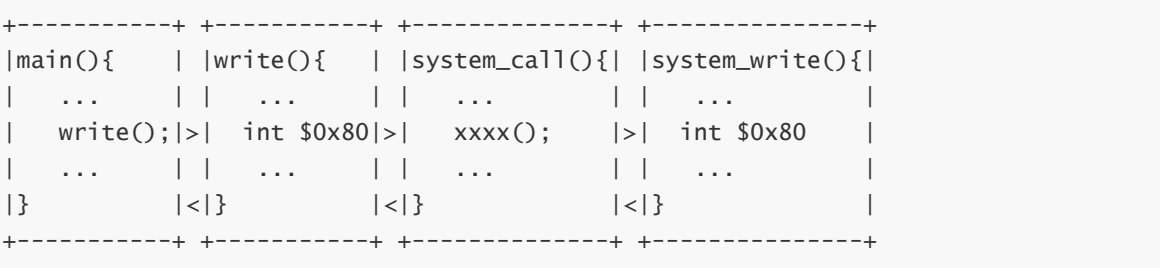
1)

如下图所示：



2)

函数调用关系如下图所示：



首先通过call指令进入`write()`函数，在`write()`函数内执行一段包含自陷指令`int $0x80`的指令序列. 包括将`main`栈帧中的参数——字符串长度`0xe`, 指向字符串的指针, 描述符`0x1`分别送EDX, ECX, EBX, 将调用号`0x4`送EAX.

执行完`int $0x80`后，从用户态陷入内核态，调用系统处理程序`system_call()`. 根据调用号`0x4`跳转到相应的系统调用服务例程`sys_write()`执行，完成将字符串写入文件的功能。字符串首地址由ECX给出，长度由EDX指出，文件描述符由EBX给出。 `system_call()`结束后，从内核返回的参数存放在EAX中。

3)

本题给出的程序设计的便捷性和灵活性更好，由于采用高级语言实现，只要改变`write()`函数中的不同实参，就可以完成不同的功能，而上一题的汇编实现方式则需要 重新编写指令。

上一题实现方式的执行性能更好。汇编语言实现省去了高级语言中的大量函数调用，执行时间更短。

5

1)

①因为`hello.c`中使用了C标准库函数`printf()`.

②因为`stdio.h`头文件中有`printf()`函数的声明，`printf()`是C标准库函数. 所以在预处理时编译器能得到`printf()`原型的声明，在链接器链接时从C标准库`libc.a`或`libc.so`中得到`printf()`的信息。

2)

将`hello.c`预处理、编译、汇编、链接生成可执行文件`hello`, 然后启动`hello`程序。

预处理过程主要执行宏展开；编译是将预处理后的文件编译生成汇编语言程序`.s`文件；汇编是将汇编语言程序变为可重定位机器语言目标代码`.o`文件；链接是将多个可重定位的机器语言目标代码以及库函数链接起来，生成可执行文件。

3)

`printf()`函数默认将输出内容输出到标准输出`stdout`上.

4)

①`48 65 6c 6c 6f 2c 77 6f 72 6c 64 0a 00`

②`.rodata`节

③只读代码段

5)

①`libc.a`

②只读代码段

③共享库映射区

6)

`write`函数内的参数为`write(1, "Hello, world.\n", 14);`

注释如下：

```
push    %ebx                        #被调用者EBX入栈
mov     0x10(%esp), %edx            #字符串长度(14)送EDX
mov     0xc(%esp), %ecx             #字符串指针(首地址)送ECX
mov     0x8(%esp), %ebx             #文件描述符1送EBX
mov     $0x4, %eax                  #调用号4送EAX
int     $0x80                       #自陷指令，系统调用入口
pop     %ebx                        #恢复EBX旧值
cmp     $0xfffff001, %eax           #系统调用返回值>=0xfffff001?
jae     8051910<__syscall_error>    #大于等于则调用__syscall_error处理错误
ret                                 #返回调用write()的过程
```

最大错误号为`0xfffff001`，其对应十进制值是-4095，取负得到4095.

7)

本题的便捷性和灵活性最好。只要在支持C语言环境下的系统都能执行。第四题中使用的`write()`函数只能在支持`write()`系统调用的系统(类UNIX系统)中执行。第三题汇编实现形式的缺点在4-(3)中已给出解释，这里不再赘述。

第三题的汇编方式实现的执行性能最好，执行时间也最短。因为省去了大量的函数调用。

6

1)

内核的设备驱动程序层

2)

内核的设备无关软件层

3)

用户`I/O`软件层

4)

内核的设备驱动程序层和中断服务程序层

5)

内核的设备驱动程序层和中断服务程序层

8

达到最快打印速度时，打印速度为

$$50 \times 80 \times 6 / 60 = 400 \text{字}/s$$

该CPU采用中断控制`I/O`打印400字所需时间为

$$1000 \times 400 \times 1000\,(ms) \,/\, 500M = 0.8\,ms$$

显然`0.8ms << 1s`，所以可以采用终端控制`I/O`方式进行字符打印输出。