

1. Operating System Concept Chapter 4 Exercises :

4.8

(1) 在单独的线程中服务每个请求的Web服务器。

(2) 交互式GUI程序（例如调试器），其中一个线程用于监视用户输入，另一个线程表示运行中。

4.9

需要响应性，如交互程序，交互程序需要进行多个用户操作同时处理。需要资源共享时。需要节约经济，节省内容空间时。需要具有可伸缩性时。

4.10

a和d 不共享，b和c共享

多线程进程的线程共享堆内存和全局变量，每个线程都有其单独的寄存器值集和单独的堆栈

4.17

a.除去自身一共创建了5个子进程

b.一共创建了两个线程。

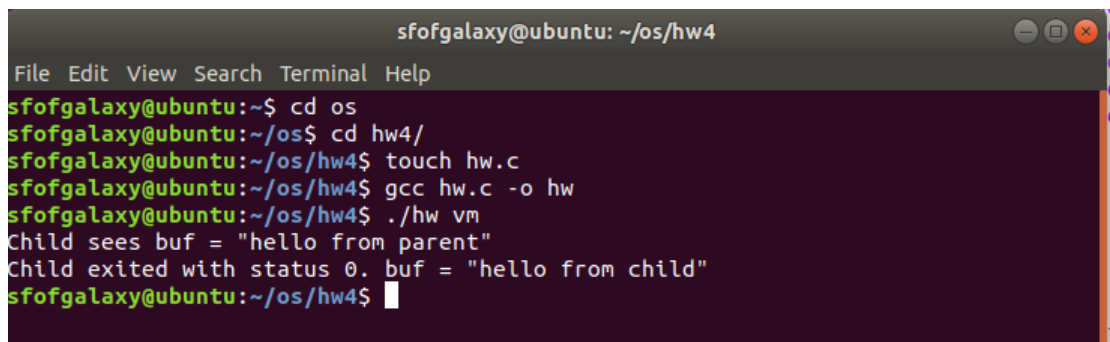
4.19

`/*LINE C*/CHILD: value = 5`

`/*LINE P*/PARENT: value = 0`

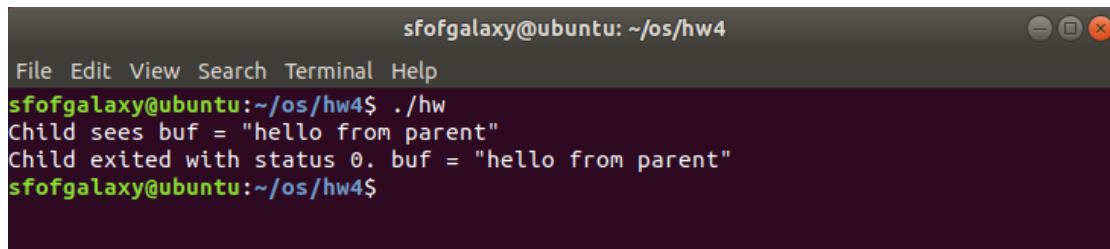
2. 运行程序：

有 vm：



```
sfofgalaxy@ubuntu: ~/os/hw4
File Edit View Search Terminal Help
sfofgalaxy@ubuntu:~$ cd os
sfofgalaxy@ubuntu:~/os$ cd hw4/
sfofgalaxy@ubuntu:~/os/hw4$ touch hw.c
sfofgalaxy@ubuntu:~/os/hw4$ gcc hw.c -o hw
sfofgalaxy@ubuntu:~/os/hw4$ ./hw vm
Child sees buf = "hello from parent"
Child exited with status 0. buf = "hello from child"
sfofgalaxy@ubuntu:~/os/hw4$
```

没有 vm：



```
sfofgalaxy@ubuntu: ~/os/hw4
File Edit View Search Terminal Help
sfofgalaxy@ubuntu:~/os/hw4$ ./hw
Child sees buf = "hello from parent"
Child exited with status 0. buf = "hello from parent"
sfofgalaxy@ubuntu:~/os/hw4$
```

原因：

如果 flag 与该 CLONE_VM 按位或则会设置为子进程与父进程运行于相同的内存空间。因此当存在 vm 时，运行相同的内存空间，则 child 在其内部改变 buf 从“hello from parent”到“hello from child”后，如果父进程再输出 buf 的时候，buf 则为“child”，因此在父进程输出时会变为“buf = "hello from child"”。如果没有 CLONE_VM，则不运行于相同的内存空间，此时使用 clone 时，仅仅复制 buf 一份，因此输出 buf 时还是“buf = "hello from parent"”