HW1

彭子帆　3170105860

1.

1-14

An interrupt is a hardware-generated change of flow within the system. An interrupt handler is summoned to deal with the cause of the interrupt; control is then returned to the interrupted context and instruction. A trap is a software-generated interrupt. An interrupt can be used to signal the completion of an I/O to obviate the need for device polling. A trap can be used to call operating system routines or to catch arithmetic errors.

1-17

An operating system for a machine of this type would need to remain in control (or monitor mode) at all times. This could be accomplished by two methods:

a. Software interpretation of all user programs (like some BASIC, Java, and LISP systems, for example). The software interpreter would provide, in software, what the hardware does not provide.

b. Require that all programs be written in high-level languages so that all object code is compilerproduced. The compiler would generate (either in-line or by function calls) the protection checks that the hardware is missing.

1-19

Slowest　f　c　a　e　d　g　b　fastest

1-22

The processor could keep track of what locations are associated with each process and limit access to locations that are outside of a program's extent. Information regarding the extent of a program's memory could be maintained by using base and limits registers and by performing a check for every memory access.

2.

截图如下:

```
sfofgalaxy@ubuntu:~$ cat /etc/issue
Ubuntu 18.04.3 LTS \n \l

sfofgalaxy@ubuntu:~$ cat /proc/version
Linux version 5.0.0-23-generic (buildd@lgw01-amd64-030) (gcc version 7.4.0 (Ubu
ntu 7.4.0-1ubuntu1~18.04.1)) #24~18.04.1-Ubuntu SMP Mon Jul 29 16:12:28 UTC 201
9
```

3.

a) 截图:

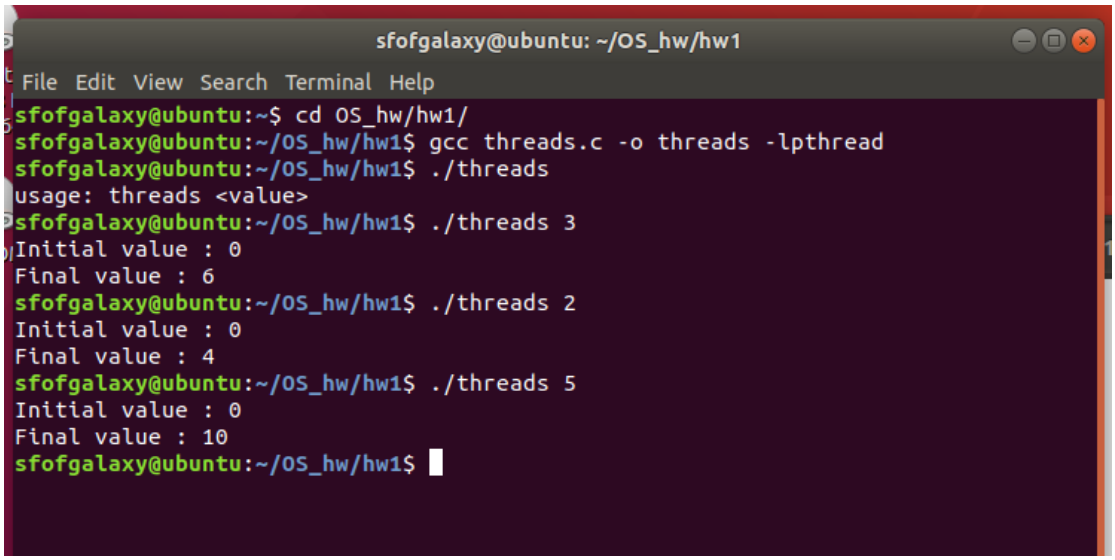
Cpu:

```
sfofgalaxy@ubuntu: ~/OS_hw/hw1
File  Edit  View  Search  Terminal  Help
sfofgalaxy@ubuntu:~$ cd OS_hw/hw1/
sfofgalaxy@ubuntu:~/OS_hw/hw1$ gcc cpu.c -o cpu -lpthread
sfofgalaxy@ubuntu:~/OS_hw/hw1$ ./cpu OS
OS
OS
OS
OS
OS
OS
^C
sfofgalaxy@ubuntu:~/OS_hw/hw1$
```

Memory:

```
sfofgalaxy@ubuntu: ~/OS_hw/hw1
File  Edit  View  Search  Terminal  Help
sfofgalaxy@ubuntu:~$ cd OS_hw/hw1/
sfofgalaxy@ubuntu:~/OS_hw/hw1$ gcc mem.c -o mem -lpthread
mem.c: In function 'main':
mem.c:11:58: warning: cast from pointer to integer of different size [-Wpointer-
to-int-cast]
     printf("(%d) memory address of p: %08x\n", getpid(), (unsigned) p);
                                                           ^
sfofgalaxy@ubuntu:~/OS_hw/hw1$ ./mem
(4015) memory address of p: 9da70260
(4015) p: 1
(4015) p: 2
(4015) p: 3
(4015) p: 4
(4015) p: 5
(4015) p: 6
(4015) p: 7
(4015) p: 8
(4015) p: 9
(4015) p: 10
^C
sfofgalaxy@ubuntu:~/OS_hw/hw1$
```

Threads:



b) 学会了什么

1. 通过这些文件，学会了如何组织这些文件，可以并且如何进行编译、运行，并且链接不是标准库中的库。
2. 通过 cpu.c 学会了 main 函数中的参数的含义，运行时所需参数。
3. 通过 Common.h 中对于 Spin 函数的定义明白了该间隔时间的函数如何编写。
4. 通过 mem.c 分配内存以及进程分配内存的 ID
5. 通过 treads.c 明白一个处理器可以创建多个线程

c) 和书上一致. 程序运行和书中的对比获得的地址值不同，ASLR 把堆、栈、共享库映射等线性区布局随机化，所以每次运行程序所获得的地址值会是不同的，是一种针对缓冲区溢出的安全保护技术。