

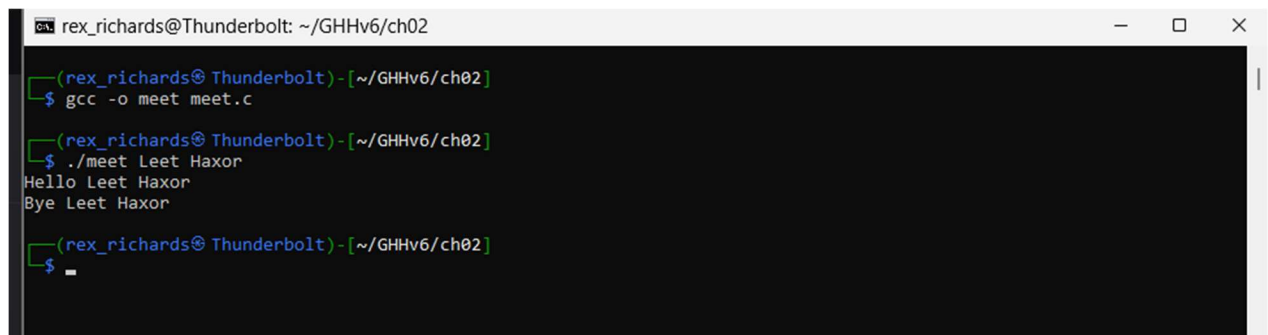
Thor Pilegaard

1/26/2025

Cybersecurity I

Assignment A

2-6.



```
rex_richards@Thunderbolt: ~/GHHv6/ch02
(rex_richards@ Thunderbolt) - [~/GHHv6/ch02]
$ gcc -o meet meet.c
(rex_richards@ Thunderbolt) - [~/GHHv6/ch02]
$ ./meet Leet Haxor
Hello Leet Haxor
Bye Leet Haxor
(rex_richards@ Thunderbolt) - [~/GHHv6/ch02]
$ _
```

2-8.

```
rex_richards@Thunderbolt: ~/GHHv6/ch02
(rex_richards@Thunderbolt)-[~/GHHv6/ch02]
$ cat ./hello.asm
section .data                ; section declaration
msg db "Hello, haxor!",0xa   ; our string with a carriage return
len equ $ - msg             ; length of our string, $ means here
section .text                ; mandatory section declaration
global _start                ; export the entry point to the ELF linker or
                             ; loaders conventionally recognize
                             ; _start as their entry point

_start:

                             ; now, write our string to stdout
                             ; notice how arguments are loaded in reverse
                             ; third argument (message length)
mov     rdx,len
mov     rcx,msg
mov     rbx,1                ; second argument (pointer to message to write)
mov     rax,4                ; load first argument (file handle (stdout))
int     0x80                 ; system call number (4=sys_write)
                             ; call kernel interrupt and exit
mov     rbx,0                ; load first syscall argument (exit code)
mov     rax,1                ; system call number (1=sys_exit)
int     0x80                 ; call kernel interrupt and exit

(rex_richards@Thunderbolt)-[~/GHHv6/ch02]
$ nasm -felf64 hello.asm

(rex_richards@Thunderbolt)-[~/GHHv6/ch02]
$ ld -s -o hello hello.o

(rex_richards@Thunderbolt)-[~/GHHv6/ch02]
$ ./hello
Hello, haxor!

(rex_richards@Thunderbolt)-[~/GHHv6/ch02]
$ _
```

2-9.

```
rex_richards@Thunderbolt: ~/GHHv6/ch02
(rex_richards@ Thunderbolt) - [~/GHHv6/ch02]
$ sudo apt-get update
Hit:1 http://http.kali.org/kali kali-last-snapshot InRelease
Reading package lists... Done

(rex_richards@ Thunderbolt) - [~/GHHv6/ch02]
$ sudo apt install gdb

Upgrading:
  libc-bin      libc6      libelf1t64    libsystemd-shared linux-base    systemd      udev
  libc-dev-bin  libc6-dev  libpam-systemd libsystemd0      locales-all  systemd-dev
  libc-l10n     libdw1t64 libssl3t64    libudev1         openssl       systemd-sysv

Installing:
  gdb

Installing dependencies:
  libbabeltrace1      libdebuginfod1t64  libpython3.12-minimal  libsource-highlight-common  openssl-provider-legacy
  libc6-dbg           libipt2            libpython3.12-stdlib  libsource-highlight4t64     systemd-cryptsetup
  libdebuginfod-common  libns12           libpython3.12t64      linux-sysctl-defaults

Suggested packages:
  gdb-doc gdbserver

Summary:
  Upgrading: 19, Installing: 15, Removing: 0, Not Upgrading: 1078
  Download size: 47.4 MB
  Space needed: 61.0 MB / 1,022 GB available

Continue? [Y/n] y
Get:1 http://kali.download/kali kali-last-snapshot/main amd64 libc-l10n all 2.40-3 [724 kB]
Get:8 http://mirrors.jevincanders.net/kali kali-last-snapshot/main amd64 systemd amd64 256.6-1 [3,488 kB]
Get:10 http://kali.mirror.rafal.ca/kali kali-last-snapshot/main amd64 libc6-dev amd64 2.40-3 [1,956 kB]
Get:2 http://mirror.math.princeton.edu/pub/kali kali-last-snapshot/main amd64 libsystemd-shared amd64 256.6-1 [2,061 kB]
Get:3 http://kali.download/kali kali-last-snapshot/main amd64 libsystemd0 amd64 256.6-1 [381 kB]
Get:4 http://kali.download/kali kali-last-snapshot/main amd64 libpam-systemd amd64 256.6-1 [278 kB]
Get:5 http://kali.download/kali kali-last-snapshot/main amd64 systemd-dev all 256.6-1 [67.9 kB]
Get:6 http://kali.download/kali kali-last-snapshot/main amd64 locales-all amd64 2.40-3 [11.1 MB]
Get:7 http://mirror.leitecastro.com/kali kali-last-snapshot/main amd64 libc6 amd64 2.40-3 [2,807 kB]
```

rex\_richards@Thunderbolt: ~/GHHv6/ch02

Processing triggers for shared-mime-info (2.4-4) ...  
Processing triggers for kali-menu (2023.4.7) ...

```
(rex_richards@Thunderbolt)-[~/GHHv6/ch02]
$ gcc -ggdb -mpreferred-stack-boundary=4 -fno-stack-protector -o meet meet.c
```

```
(rex_richards@Thunderbolt)-[~/GHHv6/ch02]
$ gdb -q meet
Reading symbols from meet...
(gdb) run 1337 Haxor
Starting program: /home/rex_richards/GHHv6/ch02/meet 1337 Haxor
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Hello 1337 Haxor
Bye 1337 Haxor
[Inferior 1 (process 1043) exited normally]
```

```
(gdb) b main
Breakpoint 1 at 0x555555551b2: file meet.c, line 10.
(gdb) run 1337 Haxor
Starting program: /home/rex_richards/GHHv6/ch02/meet 1337 Haxor
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
```

```
Breakpoint 1, main (argc=3, argv=0x7fffffffdf8) at meet.c:10
10     greeting(argv[1], argv[2]);    // call function, pass title & name
```

```
(gdb) n
Hello 1337 Haxor
11     printf("Bye %s %s\n", argv[1], argv[2]); // say "bye"
```

```
(gdb) n
Bye 1337 Haxor
12     }                                // exit program
```

```
(gdb) p argv[1]
$1 = 0x7fffffff2a8 "1337"
```

```
(gdb) p argv[2]
$2 = 0x7fffffff2ad "Haxor"
```

```
(gdb) p argc
$3 = 3
```

```
(gdb) info b
Num    Type          Disp Enb Address          What
1      breakpoint     keep y   0x0000555555551b2 in main at meet.c:10
breakpoint already hit 1 time
```

```
(gdb) info reg
rax                0x0                0
rbx                0x7fffffffdf8     140737488347112
rcx                0x0                0
rdx                0x0                0
rsi                0x5555555592a0    93824992252576
rdi                0x7fffffffdfce0 140737488346336
rbp                0x7fffffffdfded0 0x7fffffffdfded0
rsp                0x7fffffffdfdec0 0x7fffffffdfdec0
r8                 0x73               115
r9                 0xffffffff         4294967295
r10                0x0                0
r11                0x202              514
r12                0x0                0
r13                0x7fffffffef008    140737488347144
r14                0x7ffff7ffd000     140737354125312
r15                0x555555557dd8     93824992247256
rip                0x555555555205     0x555555555205 <main+98>
eflags             0x206              [ PF IF ]
cs                 0x33               51
ss                 0x2b               43
```

```
rex_richards@Thunderbolt: ~/GHHv6/ch02
[Inferior 1 (process 1043) exited normally]
(gdb) b main
Breakpoint 1 at 0x555555551b2: file meet.c, line 10.
(gdb) run 1337 Haxor
Starting program: /home/rex_richards/GHHv6/ch02/meet 1337 Haxor
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main (argc=3, argv=0x7fffffffdf8) at meet.c:10
10      greeting(argv[1], argv[2]);           // call function, pass title & name
(gdb) n
Hello 1337 Haxor
11      printf("Bye %s %s\n", argv[1], argv[2]); // say "bye"
(gdb) n
Bye 1337 Haxor
12      }                                     // exit program
(gdb) p argv[1]
$1 = 0x7fffffff2a8 "1337"
(gdb) p argv[2]
$2 = 0x7fffffff2ad "Haxor"
(gdb) p argc
$3 = 3
(gdb) info b
Num      Type      Disp Enb Address      What
1        breakpoint keep y 0x0000555555551b2 in main at meet.c:10
        breakpoint already hit 1 time
(gdb) info reg
rax      0x0      0
rbx      0x7fffffffdf8 140737488347112
rcx      0x0      0
rdx      0x0      0
rsi      0x5555555592a0 93824992252576
rdi      0x7fffffffdfce0 140737488346336
rbp      0x7fffffffdfded0 0x7fffffffdfded0
rsp      0x7fffffffdfdec0 0x7fffffffdfdec0
r8       0x73      115
r9       0xfffffffff 4294967295
r10      0x0      0
r11      0x202     514
r12      0x0      0
r13      0x7fffffffef008 140737488347144
r14      0x7ffff7ffdf000 140737354125312
r15      0x555555557dd8 93824992247256
rip      0x555555555205 0x555555555205 <main+98>
eflags   0x206     [ PF IF ]
cs       0x33     51
ss       0x2b     43
ds       0x0      0
es       0x0      0
fs       0x0      0
gs       0x0      0
fs_base  0x7ffff7db7740 140737351743296
gs_base  0x0      0
(gdb) quit
A debugging session is active.

        Inferior 1 [process 1046] will be killed.

Quit anyway? (y or n) y
(rex_richards@Thunderbolt)-[~/GHHv6/ch02]
$
```

2-10.

rex\_richards@Thunderbolt: ~/GHHv6/ch02

(rex\_richards@ Thunderbolt) - [~/GHHv6/ch02]

\$ gdb -q meet

Reading symbols from meet...

(gdb) disassemble greeting

Dump of assembler code for function greeting:

```
0x0000000000001149 <+0>:    push    %rbp
0x000000000000114a <+1>:    mov     %rsp,%rbp
0x000000000000114d <+4>:    sub     $0x1a0,%rsp
0x0000000000001154 <+11>:   mov     %rdi,-0x198(%rbp)
0x000000000000115b <+18>:   mov     %rsi,-0x1a0(%rbp)
0x0000000000001162 <+25>:   mov     -0x1a0(%rbp),%rdx
0x0000000000001169 <+32>:   lea     -0x190(%rbp),%rax
0x0000000000001170 <+39>:   mov     %rdx,%rsi
0x0000000000001173 <+42>:   mov     %rax,%rdi
0x0000000000001176 <+45>:   call    0x1030 <strcpy@plt>
0x000000000000117b <+50>:   lea     -0x190(%rbp),%rdx
0x0000000000001182 <+57>:   mov     -0x198(%rbp),%rax
0x0000000000001189 <+64>:   mov     %rax,%rsi
0x000000000000118c <+67>:   lea     0xe71(%rip),%rax      # 0x2004
0x0000000000001193 <+74>:   mov     %rax,%rdi
0x0000000000001196 <+77>:   mov     $0x0,%eax
0x000000000000119b <+82>:   call    0x1040 <printf@plt>
0x00000000000011a0 <+87>:   nop
0x00000000000011a1 <+88>:   leave
0x00000000000011a2 <+89>:   ret
```

End of assembler dump.

(gdb) set disassembly-flavor intel

(gdb) disassemble greeting

Dump of assembler code for function greeting:

```
0x0000000000001149 <+0>:    push    rbp
0x000000000000114a <+1>:    mov     rbp,rsp
0x000000000000114d <+4>:    sub     rsp,0x1a0
0x0000000000001154 <+11>:   mov     QWORD PTR [rbp-0x198],rdi
0x000000000000115b <+18>:   mov     QWORD PTR [rbp-0x1a0],rsi
0x0000000000001162 <+25>:   mov     rdx,QWORD PTR [rbp-0x1a0]
0x0000000000001169 <+32>:   lea     rax,[rbp-0x190]
0x0000000000001170 <+39>:   mov     rsi,rdx
0x0000000000001173 <+42>:   mov     rdi,rax
0x0000000000001176 <+45>:   call    0x1030 <strcpy@plt>
0x000000000000117b <+50>:   lea     rdx,[rbp-0x190]
0x0000000000001182 <+57>:   mov     rax,QWORD PTR [rbp-0x198]
0x0000000000001189 <+64>:   mov     rsi,rax
0x000000000000118c <+67>:   lea     rax,[rip+0xe71]      # 0x2004
0x0000000000001193 <+74>:   mov     rdi,rax
0x0000000000001196 <+77>:   mov     eax,0x0
0x000000000000119b <+82>:   call    0x1040 <printf@plt>
0x00000000000011a0 <+87>:   nop
0x00000000000011a1 <+88>:   leave
0x00000000000011a2 <+89>:   ret
```

End of assembler dump.

(gdb) quit

(rex\_richards@ Thunderbolt) - [~/GHHv6/ch02]

\$