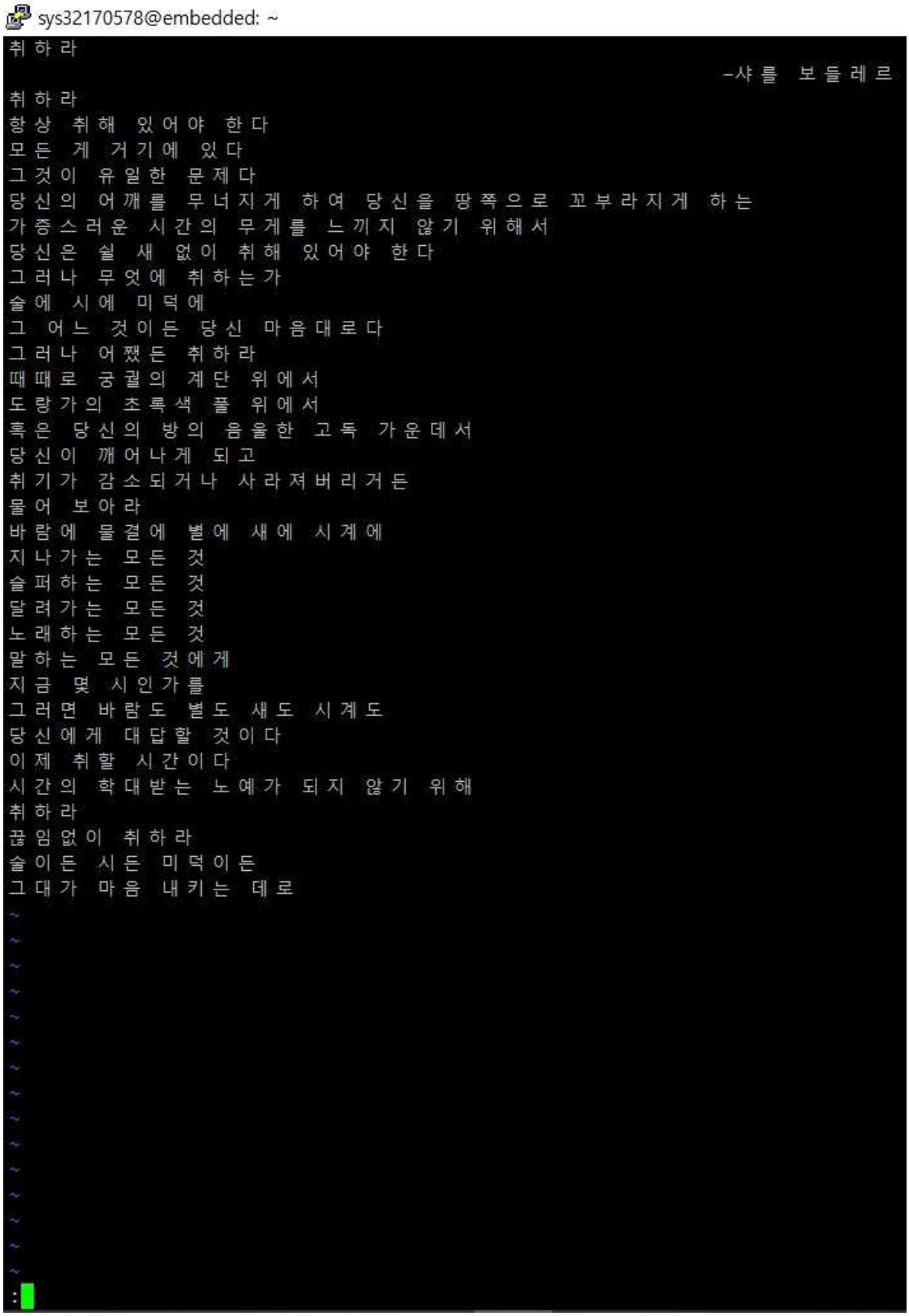
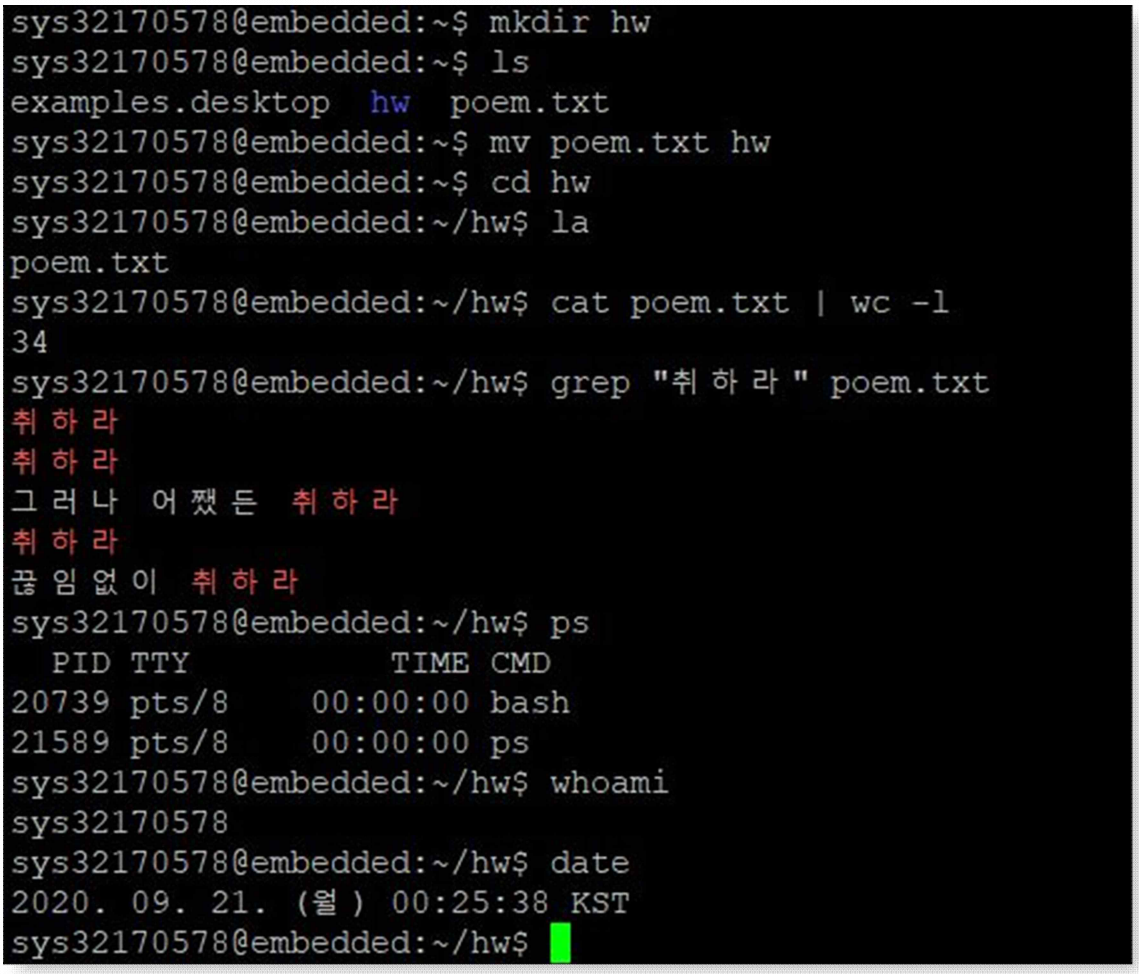
과제 2:리눅스 명령어 실습

시스템 프로그래밍 7분반
소프트웨어학과 32170578 김산

<요구조건 1 : vi 편집기로 자신이 좋아하는 시 작성 (흥미로운 시 선택)>



<요구조건 2: 리눅스 명령어 10개 이상 실습하고 스냅샷 작성>



1. 파일을 정리하기위해 mkdir 명령을 사용하여 hw 폴더 생성
2. ls 명령을 통해 현재 디렉토리에 폴더 생성이 되었는지 확인
3. 작성한 현재디렉토리의 poem.txt 파일을 mv 명령을 통해 hw폴더로 이동
4. cd 명령을 통해 hw디렉토리로 이동
5. la 명령을 통해 파일 이동 확인
6. cat poem.txt | wc -l 명령을 통해 시가 몇줄인지 확인
7. grep 명령을 통해 poem.txt 에 “취하라” 라는 단어가 들어간 문장을 추출
8. ps명령을 통해 현재 실행되고 있는 프로세스를 확인
9. whoami 명령을 통해 사용자 id확인
10. data 명령을 통해 현재 날짜 확인

<보너스 : 컴파일 관련 명령어 실습>

- vi 편집기로 프로그램 작성

sys32170578@embedded: ~/hw

```
#include<stdio.h>

char a[4] = {124,125,126,127};
int *p = NULL;
main(){
        int i = 0;;
        for(i = 0; i < 6; i++){
                a[i]++;
                printf("%d\n",a[i]);
        }
        *p = 1;
        printf("%d\n", *p);
}
~
```

- gcc -o hello.out hello.c 명령으로 hello.out 파일 생성

sys32170578@embedded: ~/hw

```
sys32170578@embedded:~/hw$ gcc -o hello.out hello.c
sys32170578@embedded:~/hw$ ./hello.out
125
126
127
-128
1
1
Segmentation fault (core dumped)
```

- Segmentation fault 오류 발생
- 자료형의 범위를 초과하는 경우에서는 오류가 발생하지 않는다.

- gdb 디버깅

```
sys32170578@embedded:~/hw$ gdb hello.out
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/li
This is free software: you are free to change and redistribut
There is NO WARRANTY, to the extent permitted by law.  Type "
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
Reading symbols from hello.out...done.
(gdb) run
Starting program: /home/7-class/sys32170578/hw/hello.out
125
126
127
-128
1
1

Program received signal SIGSEGV, Segmentation fault.
0x080483e3 in main () at hello.c:11
11              *p = 1;
(gdb) list
6               int i = 0;;
7               for(i = 0; i < 6; i++){
8                       a[i]++;
9                       printf("%d\n",a[i]);
10              }
11              *p = 1;
12              printf("%d\n", *p);
13      }
(gdb) break 10
Breakpoint 1 at 0x80483de: file hello.c, line 10.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/7-class/sys32170578/hw/hello.out
```

- 10번째 줄을 break point 로 설정하여 오류 발생문장 확인

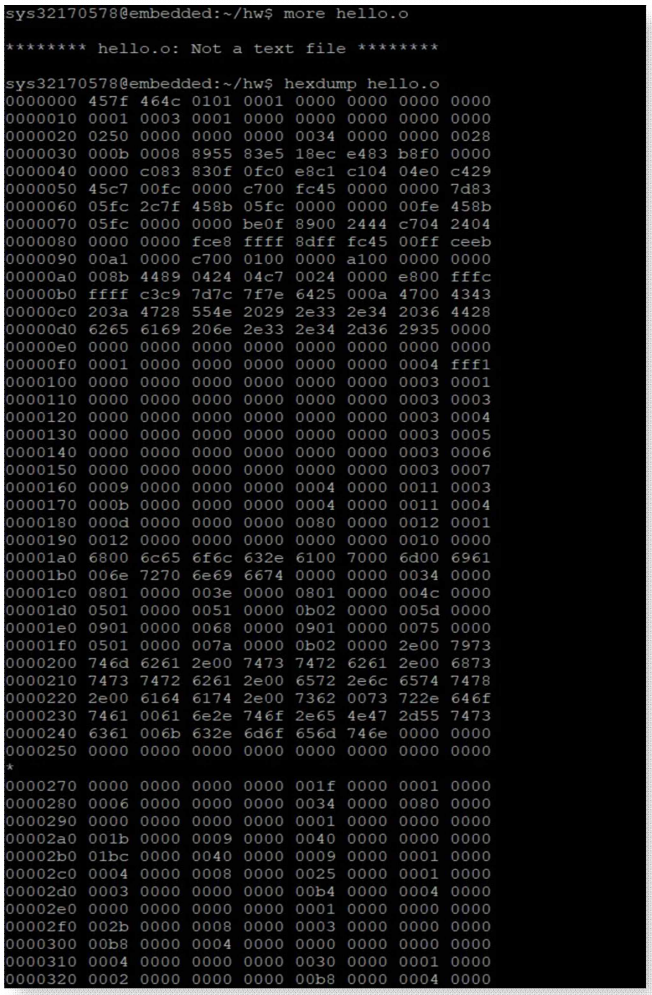
```
125
126
127
-128
1
1

Breakpoint 1, main () at hello.c:11
11              *p = 1;
(gdb) n

Program received signal SIGSEGV, Segmentation fault.
0x080483e3 in main () at hello.c:11
11              *p = 1;
(gdb)
```

- more 명령어로 어셈블리 파일 hello.s 내용 확인, as -o 명령어를 사용하여 바이너리 파일 hello.o 생성

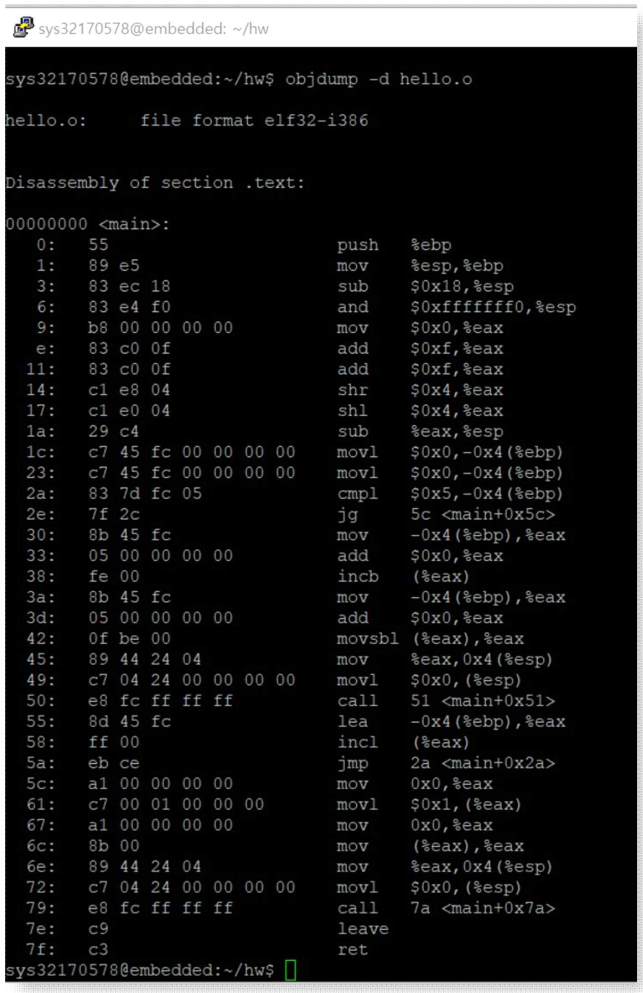
sys32170578@embedded: ~/hw

```
sys32170578@embedded:~/hw$ as -o hello.o hello.s
sys32170578@embedded:~/hw$ ls
a.out  hello.c  hello.o  hello.out  hello.s  poem.txt
sys32170578@embedded:~/hw$ more hello.s
        .file   "hello.c"
.globl a
        .data
        .type   a, @object
        .size   a, 4
a:
        .byte   124
        .byte   125
        .byte   126
        .byte   127
.globl p
        .bss
        .align 4
        .type   p, @object
        .size   p, 4
p:
        .zero   4
        .section        .rodata
.LC0:
        .string "%d\n"
        .text
.globl main
        .type   main, @function
main:
        pushl   %ebp
        movl    %esp, %ebp
        subl    $24, %esp
        andl    $-16, %esp
        movl    $0, %eax
        addl    $15, %eax
        addl    $15, %eax
        shrl    $4, %eax
        sall    $4, %eax
        subl    %eax, %esp
        movl    $0, -4(%ebp)
        movl    $0, -4(%ebp)
.L2:
        cmpl    $5, -4(%ebp)
        jg      .L3
        movl    -4(%ebp), %eax
        addl    $a, %eax
        incb    (%eax)
        movl    -4(%ebp), %eax
        addl    $a, %eax
        movsbl  (%eax),%eax
        movl    %eax, 4(%esp)
        movl    $.LC0, (%esp)
        call    printf
        leal    -4(%ebp), %eax
        incl    (%eax)
        jmp     .L2
.L3:
```

- hexdump를 통해 바이너리 파일 열람, objdump로 파일 분석



```
sys32170578@embedded:~/hw$ more hello.o

******** hello.o: Not a text file ********

sys32170578@embedded:~/hw$ hexdump hello.o
0000000 457f 464c 0101 0001 0000 0000 0000 0000
0000010 0001 0003 0001 0000 0000 0000 0000 0000
0000020 0250 0000 0000 0000 0034 0000 0000 0028
0000030 000b 0008 8955 83e5 18ec e483 b8f0 0000
0000040 0000 c083 830f 0fc0 e8c1 c104 04e0 c429
0000050 45c7 00fc 0000 c700 fc45 0000 0000 7d83
0000060 05fc 2c7f 458b 05fc 0000 0000 00fe 458b
0000070 05fc 0000 0000 be0f 8900 2444 c704 2404
0000080 0000 0000 fce8 ffff 8dff fc45 00ff ceeb
0000090 00a1 0000 c700 0100 0000 a100 0000 0000
00000a0 008b 4489 0424 04c7 0024 0000 e800 fffc
00000b0 ffff c3c9 7d7c 7f7e 6425 000a 4700 4343
00000c0 203a 4728 554e 2029 2e33 2e34 2036 4428
00000d0 6265 6169 206e 2e33 2e34 2d36 2935 0000
00000e0 0000 0000 0000 0000 0000 0000 0000 0000
00000f0 0001 0000 0000 0000 0000 0000 0004 fff1
0000100 0000 0000 0000 0000 0000 0000 0003 0001
0000110 0000 0000 0000 0000 0000 0000 0003 0003
0000120 0000 0000 0000 0000 0000 0000 0003 0004
0000130 0000 0000 0000 0000 0000 0000 0003 0005
0000140 0000 0000 0000 0000 0000 0000 0003 0006
0000150 0000 0000 0000 0000 0000 0000 0003 0007
0000160 0009 0000 0000 0000 0004 0000 0011 0003
0000170 000b 0000 0000 0000 0004 0000 0011 0004
0000180 000d 0000 0000 0000 0080 0000 0012 0001
0000190 0012 0000 0000 0000 0000 0000 0010 0000
00001a0 6800 6c65 6f6c 632e 6100 7000 6d00 6961
00001b0 006e 7270 6e69 6674 0000 0000 0034 0000
00001c0 0801 0000 003e 0000 0801 0000 004c 0000
00001d0 0501 0000 0051 0000 0b02 0000 005d 0000
00001e0 0901 0000 0068 0000 0901 0000 0075 0000
00001f0 0501 0000 007a 0000 0b02 0000 2e00 7973
0000200 746d 6261 2e00 7473 7472 6261 2e00 6873
0000210 7473 7472 6261 2e00 6572 2e6c 6574 7478
0000220 2e00 6164 6174 2e00 7362 0073 722e 646f
0000230 7461 0061 6e2e 746f 2e65 4e47 2d55 7473
0000240 6361 006b 632e 6d6f 656d 746e 0000 0000
0000250 0000 0000 0000 0000 0000 0000 0000 0000
*
0000270 0000 0000 0000 0000 001f 0000 0001 0000
0000280 0006 0000 0000 0000 0034 0000 0080 0000
0000290 0000 0000 0000 0000 0001 0000 0000 0000
00002a0 001b 0000 0009 0000 0040 0000 0000 0000
00002b0 01bc 0000 0040 0000 0009 0000 0001 0000
00002c0 0004 0000 0008 0000 0025 0000 0001 0000
00002d0 0003 0000 0000 0000 00b4 0000 0004 0000
00002e0 0000 0000 0000 0000 0001 0000 0000 0000
00002f0 002b 0000 0008 0000 0003 0000 0000 0000
0000300 00b8 0000 0004 0000 0000 0000 0000 0000
0000310 0004 0000 0000 0000 0030 0000 0001 0000
0000320 0002 0000 0000 0000 00b8 0000 0004 0000
```



```
sys32170578@embedded: ~/hw

sys32170578@embedded:~/hw$ objdump -d hello.o

hello.o:     file format elf32-i386


Disassembly of section .text:

00000000 <main>:
   0:   55                      push   %ebp
   1:   89 e5                   mov    %esp,%ebp
   3:   83 ec 18                sub    $0x18,%esp
   6:   83 e4 f0                and    $0xfffffff0,%esp
   9:   b8 00 00 00 00          mov    $0x0,%eax
   e:   83 c0 0f                add    $0xf,%eax
  11:   83 c0 0f                add    $0xf,%eax
  14:   c1 e8 04                shr    $0x4,%eax
  17:   c1 e0 04                shl    $0x4,%eax
  1a:   29 c4                   sub    %eax,%esp
  1c:   c7 45 fc 00 00 00 00    movl   $0x0,-0x4(%ebp)
  23:   c7 45 fc 00 00 00 00    movl   $0x0,-0x4(%ebp)
  2a:   83 7d fc 05             cmpl   $0x5,-0x4(%ebp)
  2e:   7f 2c                   jg     5c <main+0x5c>
  30:   8b 45 fc                mov    -0x4(%ebp),%eax
  33:   05 00 00 00 00          add    $0x0,%eax
  38:   fe 00                   incb   (%eax)
  3a:   8b 45 fc                mov    -0x4(%ebp),%eax
  3d:   05 00 00 00 00          add    $0x0,%eax
  42:   0f be 00                movsbl (%eax),%eax
  45:   89 44 24 04             mov    %eax,0x4(%esp)
  49:   c7 04 24 00 00 00 00    movl   $0x0,(%esp)
  50:   e8 fc ff ff ff          call   51 <main+0x51>
  55:   8d 45 fc                lea    -0x4(%ebp),%eax
  58:   ff 00                   incl   (%eax)
  5a:   eb ce                   jmp    2a <main+0x2a>
  5c:   a1 00 00 00 00          mov    0x0,%eax
  61:   c7 00 01 00 00 00       movl   $0x1,(%eax)
  67:   a1 00 00 00 00          mov    0x0,%eax
  6c:   8b 00                   mov    (%eax),%eax
  6e:   89 44 24 04             mov    %eax,0x4(%esp)
  72:   c7 04 24 00 00 00 00    movl   $0x0,(%esp)
  79:   e8 fc ff ff ff          call   7a <main+0x7a>
  7e:   c9                      leave
  7f:   c3                      ret
sys32170578@embedded:~/hw$
```