

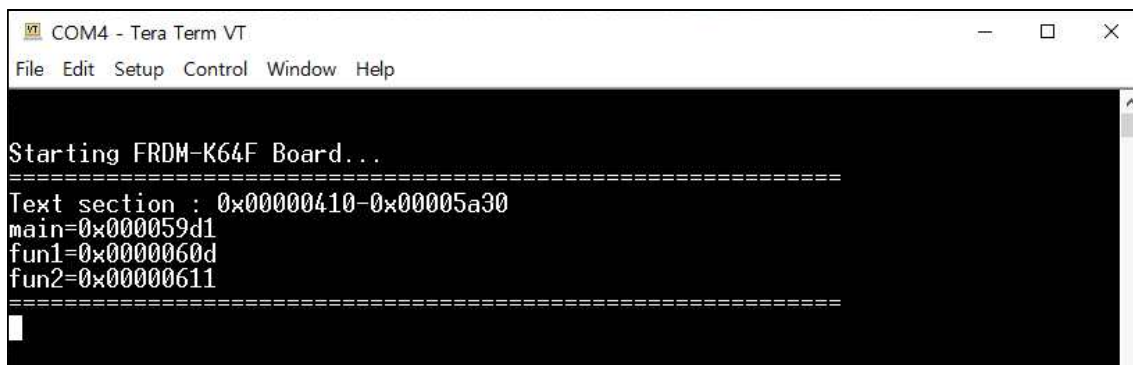
강의명 : 마이크로프로세서
실습 번호 : 2
실습 제목 : ELF Sections
학생 이름 : 이학민
학번 : 201910906

1. TEXT section 정보

1.1

```
extern unsigned int __text_start__;  
// text section 시작 주소를 출력하기 위한 변수 설정  
extern unsigned int __text_end__;  
// text section 끝 주소를 출력하기 위한 변수 설정  
  
//extern  
  
printf("Text section : 0x%08x-0x%08x\n", &__text_start__, &__text_end__);  
// text section의 처음과 끝 주소를 16진수로 나타내기 위해 %x를 이용하고 변수 앞에 &를 붙였음.  
printf("main=0x%08x\n", main); // C 프로그램 내 main 함수의 주소를 화면에 출력함  
printf("fun1=0x%08x\n", fun1); // C 프로그램 내 fun1 함수의 주소를 화면에 출력함  
printf("fun2=0x%08x\n", fun2); // C 프로그램 내 fun2 함수의 주소를 화면에 출력함  
// %08x의 의미 : 16진수 형식 8자리로 출력, 총 자릿수가 8자리 미만이면 앞자리를 모두 0으로 채움.
```

1.2



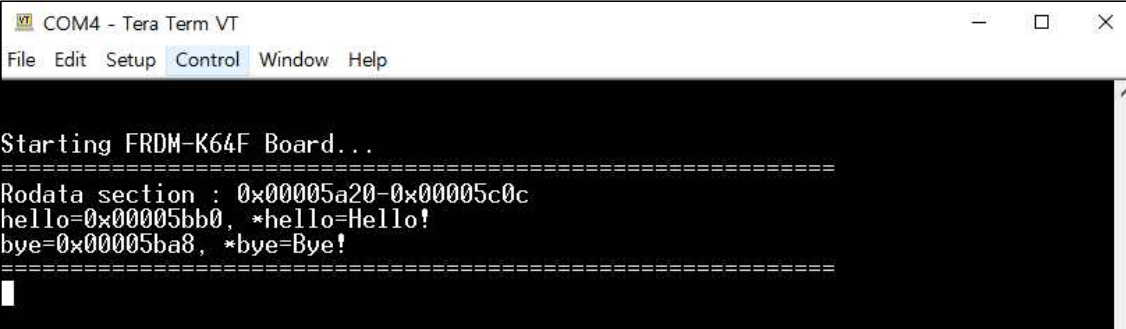
```
COM4 - Tera Term VT  
File Edit Setup Control Window Help  
Starting FRDM-K64F Board...  
=====  
Text section : 0x00000410-0x00005a30  
main=0x000059d1  
fun1=0x0000060d  
fun2=0x00000611  
=====
```

2. RODATA section 정보

2.1

```
extern unsigned int __rodata_start__;  
// rodata section 시작 주소를 출력하기 위한 변수 설정  
extern unsigned int __rodata_end__;  
// rodata section 끝 주소를 출력하기 위한 변수 설정  
  
printf("Rodata section : 0x%08x-0x%08x\n", &__rodata_start__, &__rodata_end__);  
// rodata section의 처음과 끝 주소를 16진수로 나타내기 위해 %x를 이용하고 변수 앞에  
&를 붙였음.  
printf("hello=0x%08x, *hello=%s\n",hello, hello);  
// hello는 포인터 변수로 선언되었기 때문에 주소를 출력하기 위해 &를 붙이지 않고 그냥  
쓴다. 또한 %s를 이용하여 Hello!가 출력되게 만들었음.  
printf("bye=0x%08x, *bye=%s\n",bye, bye);  
// bye는 포인터 변수로 선언되었기 때문에 주소를 출력하기 위해 &를 붙이지 않고 그냥 쓴  
다. 또한 %s를 이용하여 Bye!가 출력되게 만들었음.
```

2.2



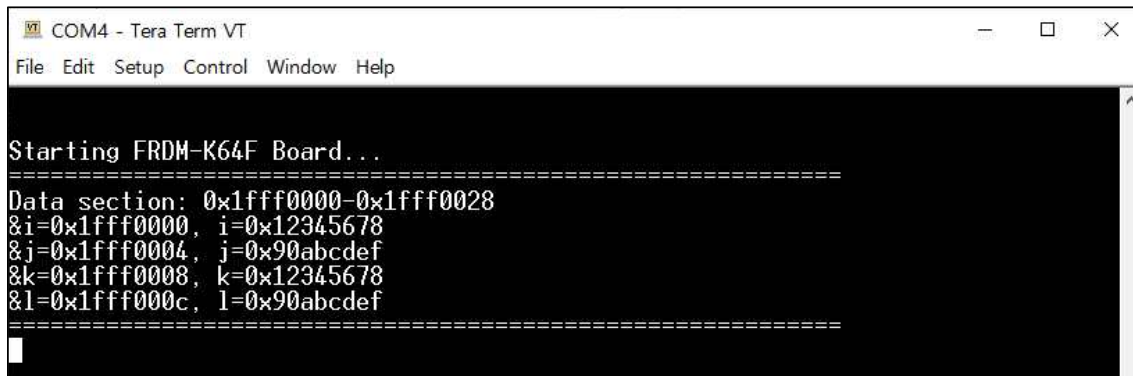
```
COM4 - Tera Term VT  
File Edit Setup Control Window Help  
Starting FRDM-K64F Board...  
=====  
Rodata section : 0x00005a20-0x00005c0c  
hello=0x00005bb0, *hello=Hello!  
bye=0x00005ba8, *bye=Bye!  
=====
```

3. DATA section 정보

3.1

```
extern unsigned int __data_start__;  
// data section 시작 주소를 출력하기 위한 변수 설정  
extern unsigned int __data_end__;  
// data section 끝 주소를 출력하기 위한 변수 설정  
  
printf("Data section: 0x%08x-0x%08x\n",&__data_start__, &__data_end__);  
// data section의 처음과 끝 주소를 16진수로 나타내기 위해 %x를 이용하고 변수 앞에 &  
를 붙였음.  
printf("&i=0x%08x, i=0x%08x\n",&i, i);  
printf("&j=0x%08x, j=0x%08x\n",&j, j);  
printf("&k=0x%08x, k=0x%08x\n",&k, k);  
printf("&l=0x%08x, l=0x%08x\n",&l, l);  
// 각 변수의 주소를 출력하기 위해 변수 앞에 &를 붙여 사용하고 각 변수값을 출력하기 위  
해서는 &를 붙이지 않은 채 사용함.
```

3.2



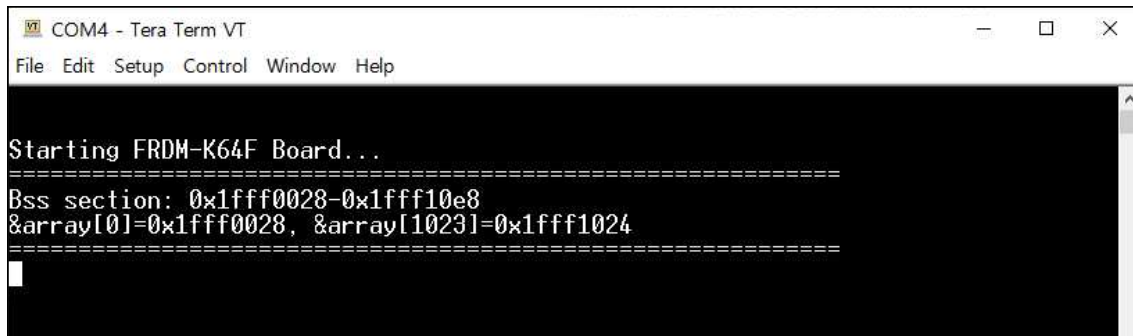
```
COM4 - Tera Term VT  
File Edit Setup Control Window Help  
  
Starting FRDM-K64F Board...  
=====  
Data section: 0x1fff0000-0x1fff0028  
&i=0x1fff0000, i=0x12345678  
&j=0x1fff0004, j=0x90abcdef  
&k=0x1fff0008, k=0x12345678  
&l=0x1fff000c, l=0x90abcdef  
=====
```

4. BSS section 정보

4.1

```
extern unsigned int __bss_start__;  
// bss section 시작 주소를 출력하기 위한 변수 설정  
extern unsigned int __bss_end__;  
// bss section 끝 주소를 출력하기 위한 변수 설정  
  
printf("Bss section: 0x%08x-0x%08x\n",&__bss_start__, &__bss_end__);  
// bss section의 처음과 끝 주소를 16진수로 나타내기 위해 %x를 이용하고 변수 앞에 &를 붙였음.  
printf("&array[0]=0x%08x, &array[1023]=0x%08x\n", &array[0], &array[1023]);  
// array[0]와 array[1023]의 주소를 출력하기 위해 변수 앞에 &를 붙여 사용하였음.
```

4.2



```
COM4 - Tera Term VT  
File Edit Setup Control Window Help  
  
Starting FRDM-K64F Board...  
=====  
Bss section: 0x1fff0028-0x1fff10e8  
&array[0]=0x1fff0028, &array[1023]=0x1fff1024  
=====
```

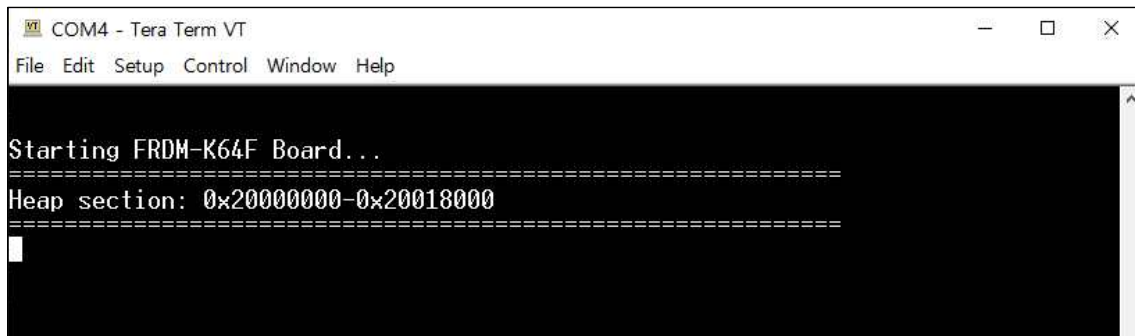
5. HEAP section 정보

5.1

```
extern unsigned int __heap_start__;  
// heap section 시작 주소를 출력하기 위한 변수 설정  
extern unsigned int __heap_end__;  
// heap section 시작 주소를 출력하기 위한 변수 설정
```

```
printf("Heap section: 0x%08x-0x%08x\n", &__heap_start__, &__heap_end__);  
// heap section의 처음과 끝 주소를 16진수로 나타내기 위해 %x를 이용하고 변수 앞에 &  
를 붙였음.
```

5.2



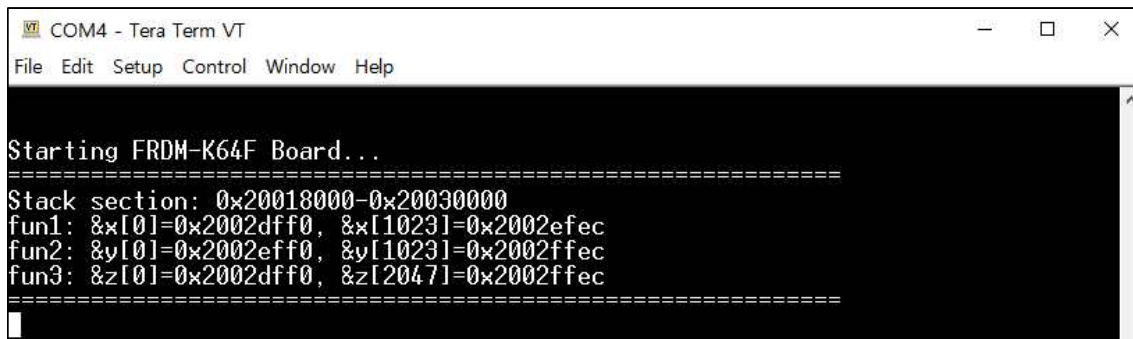
The screenshot shows a terminal window titled "COM4 - Tera Term VT". The menu bar includes "File", "Edit", "Setup", "Control", "Window", and "Help". The terminal output displays the text "Starting FRDM-K64F Board..." followed by a line of equals signs. Below this, it shows "Heap section: 0x20000000-0x20018000", also followed by a line of equals signs. A cursor is visible on the line immediately following the second line of equals signs.

6. STACK section 정보

6.1

```
extern unsigned int __stack_start__;  
// stack section 시작 주소를 출력하기 위한 변수 설정  
extern unsigned int __stack_end__;  
// stack section 시작 주소를 출력하기 위한 변수 설정  
  
printf("Stack section: 0x%08x-0x%08x\n",&__stack_start__, &__stack_end__);  
// stack section의 처음과 끝 주소를 16진수로 나타내기 위해 %x를 이용하고 변수 앞에 &  
// 를 붙였음.  
  
printf("fun1: &x[0]=0x%08x, &x[1023]=0x%08x\n", &x[0], &x[1023]);  
// x[0]와 x[1023]의 주소를 출력하기 위해 %x를 이용하고 변수 앞에 &를 붙임.  
printf("fun2: &y[0]=0x%08x, &y[1023]=0x%08x\n", &y[0], &y[1023]);  
// y[0]와 y[1023]의 주소를 출력하기 위해 %x를 이용하고 변수 앞에 &를 붙임.  
printf("fun3: &z[0]=0x%08x, &z[2047]=0x%08x\n", &z[0], &z[2047]);  
// z[0]와 z[2047]의 주소를 출력하기 위해 %x를 이용하고 변수 앞에 &를 붙임.
```

6.2



```
COM4 - Tera Term VT  
File Edit Setup Control Window Help  
  
Starting FRDM-K64F Board...  
=====  
Stack section: 0x20018000-0x20030000  
fun1: &x[0]=0x2002dff0, &x[1023]=0x2002efec  
fun2: &y[0]=0x2002eff0, &y[1023]=0x2002ffec  
fun3: &z[0]=0x2002dff0, &z[2047]=0x2002ffec  
=====
```


heap section의 시작 주소가 0x20000000인 것과 끝 주소가 0x20018000인 것을 보아 올바른 heap section을 가지고 있음을 알 수 있다. heap은 사용할수록 번지수가 증가하는 특징이 있다.

6. stack section : 0x20018000-0x20030000

stack section의 시작 주소가 0x20018000인 것이 일치한다. fun1, fun2, fun3에 존재하는 배열 x[0](0x2002dff0), x[1023](0x2002efec), y[0](0x2002eff0), y[1023](0x2002ffec), z[0](0x2002dff0), z[2047](0x2002ffec)의 주소값은 stack section 주소 범위 내에 있으므로 옳바르다. stack은 사용할수록 번지수가 감소하는 특징이 있다.

끝.