



과목명	시스템 프로그래밍
담당교수	최종무 교수님
학과	소프트웨어학과
학번	32200185
이름	곽다은
제출일자	2021.10.04

```
sys32200185@embedded: ~  
Using username "sys32200185".  
sys32200185@220.149.236.2's password:  
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.15.0-122-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
142 packages can be updated.  
2 updates are security updates.  
  
New release '18.04.6 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
sys32200185@embedded:~$ ps  
  PID TTY          TIME CMD  
16642 pts/9        00:00:00 bash  
16769 pts/9        00:00:00 ps  
sys32200185@embedded:~$ vi test.c  
sys32200185@embedded:~$ ls  
examples.desktop  test.c  
sys32200185@embedded:~$ cat test.c  
여름 것  
새 소년  
  
곧 지나갈 여름밤의 소리  
여기 어리고 새 푸른 두 눈  
들이 쉬고 내 뱀는 나의 코와  
시간을 쫓갠 입술이 먼지만 내 뿜고 있네  
지나간 사람 지나간 사랑들  
내 몸에 새겨질 삶의 타투  
나의 젊음에 네 미래를 줘  
내 망친 작품을 비웃어 줘  
음곤란해진 건  
닭이지 않는 얼룩이 었고  
음익숙해진 건  
명청한 나의 알람시계
```

1. vi test.c -> vi편집기를 사용하여 좋아하는 시를 작성하였다.
2. ls
3. cat test.c

```

sys32200185@embedded:~$ more test.c
여름 것
    새 소년

곧 지나갈 여름 밤의 소리
여기 어리고 새 푸른 두 눈
들이쉬고 내뿜는 나의 코와
시간을 쪼갬 입술이 먼지만 내뿜고 있네
지나간 사람 지나간 사랑들
내 몸에 새겨질 삶의 타투
나의 젊음에 네 미래를 줘
내 망친 작품을 비웃어줘
음곤란해진 건
닭이지 않는 얼룩이 었고
음익숙해진 건
명청한 나의 알람시계

sys32200185@embedded:~$ grep -n 얼룩 test.c | wc -l
1
sys32200185@embedded:~$ grep -n 얼룩 test.c
13:닭이지 않는 얼룩이 었고

```

4. more test.c
5. grep -n 얼룩 test.c | wc -l
6. grep -n 얼룩 test.c -> "얼룩" 키워드를 grep을 통해 test.c 내에서 찾았고, 개수와 내용을 출력시킨다.

```

sys32200185@embedded:~$ cd ../
sys32200185@embedded:/home/1-class$ ls
sys32170983 sys32180874 sys32183698 sys32200149 sys32202319 sys322040
sys32171761 sys32181070 sys32183864 sys32200185 sys32202456 sys322042
sys32172086 sys32182377 sys32184007 sys32200474 sys32203003 sys322042
sys32173594 sys32182661 sys32184731 sys32200566 sys32203279 sys322043
sys32174906 sys32182775 sys32185117 sys32200793 sys32203502 sys322047
sys32174918 sys32183163 sys32192979 sys32201063 sys32203689 sys322048
sys32180255 sys32183658 sys32193087 sys32201489 sys32203894 sys322049
sys32180527 sys32183662 sys32197252 sys32201817 sys32203931
sys32200185@embedded:/home/1-class$ cd sys32200185

```

7. cd & ls

```
sys32200185@embedded: ~  
sys32200185@embedded:~$ vi testt.c  
sys32200185@embedded:~$ cat testt.c  
#include <stdio.h>  
  
int a, b, c;  
  
main() {  
    a = 100;  
    b = 250;  
    c = b - a;  
    printf("Hello : C = %d\n", c);  
}  
sys32200185@embedded:~$ gcc -S testt.c  
sys32200185@embedded:~$ ls  
examples.desktop test.c test.s testt.c testt.s  
sys32200185@embedded:~$ more testt.s  
    .file "testt.c"  
    .section      .rodata  
.  
.LC0:  
    .string "Hello : C = %d\n"  
    .text  
.globl main  
    .type        main, @function  
main:  
    pushl        %ebp  
    movl         %esp, %ebp  
    subl         $8, %esp  
    andl         $-16, %esp  
    movl         $0, %eax  
    addl         $15, %eax  
    addl         $15, %eax  
    shrl         $4, %eax  
    sall         $4, %eax  
    subl         %eax, %esp  
    movl         $100, a  
    movl         $250, b  
    movl         a, %edx  
    movl         b, %eax  
    subl         %edx, %eax  
    movl         %eax, c  
    movl         c, %eax  
    movl         %eax, 4(%esp)  
    movl         $.LC0, (%esp)  
    call         printf
```

8. vi testt.c -> vi 편집기를 사용하여 간단한 c프로그램을 작성하였다.
9. cat testt.c
10. gcc -S testt.c -> testt.s라는 assembly파일을 생성한다.
11. more testt.s

```
sys32200185@embedded: ~  
    leave  
    ret  
    .size    main, .-main  
    .comm    a,4,4  
    .comm    b,4,4  
--More-- (83%)  
[1]+  Stopped                  more testt.s  
sys32200185@embedded:~$ ./a.out  
-bash: ./a.out: No such file or directory  
sys32200185@embedded:~$ gcc -o testt.out testt.c  
sys32200185@embedded:~$ ./testt.out  
Hello : C = 150  
sys32200185@embedded:~$ ls  
examples.desktop test.c test.s testt.c testt.out testt.s  
sys32200185@embedded:~$ as -o testt.o testt.s  
sys32200185@embedded:~$ ls  
examples.desktop test.c test.s testt.c testt.o testt.out testt.s  
sys32200185@embedded:~$ more testt.o  
***** testt.o: Not a text file *****  
  
sys32200185@embedded:~$ hexdump testt.o  
00000000 457f 464c 0101 0001 0000 0000 0000 0000  
00000010 0001 0003 0001 0000 0000 0000 0000 0000  
00000020 0240 0000 0000 0000 0034 0000 0000 0028  
00000030 000b 0008 8955 83e5 08ec e483 b8f0 0000  
00000040 0000 c083 830f 0fc0 e8c1 c104 04e0 c429  
00000050 05c7 0000 0000 0064 0000 05c7 0000 0000  
00000060 00fa 0000 158b 0000 0000 00a1 0000 2900  
00000070 a3d0 0000 0000 00a1 0000 8900 2444 c704  
00000080 2404 0000 0000 fce8 ffff c9ff 48c3 6c65  
00000090 6f6c 3a20 4320 3d20 2520 0a64 0000 4347  
000000a0 3a43 2820 4e47 2955 3320 342e 362e 2820  
000000b0 6544 6962 6e61 3320 342e 362e 352d 0029  
000000c0 0000 0000 0000 0000 0000 0000 0000 0000  
000000d0 0001 0000 0000 0000 0000 0000 0004 fff1  
000000e0 0000 0000 0000 0000 0000 0000 0003 0001  
000000f0 0000 0000 0000 0000 0000 0000 0003 0003  
00001000 0000 0000 0000 0000 0000 0000 0003 0004  
00001100 0000 0000 0000 0000 0000 0000 0003 0005  
00001200 0000 0000 0000 0000 0000 0000 0003 0006  
00001300 0000 0000 0000 0000 0000 0000 0003 0007  
00001400 0009 0000 0000 0000 0059 0000 0012 0001  
00001500 000e 0000 0004 0000 0004 0000 0011 fff2
```

12. gcc -o testt.out testt.c -> 수행가능한 testt.out을 생성한다.
13. ./testt.out -> 현재 디렉토리에서 testt.out을 실행한다.
14. as -o testt.o testt.s -> 재배치 가능한 testt.o를 생성한다.
15. hexdump testt.o -> 바이너리 파일을 읽고 출력한다.

```
sys32200185@embedded: ~
00001a0 0062 7270 6e69 6674 0000 0000 001e 0000
00001b0 0901 0000 0028 0000 0a01 0000 0032 0000
00001c0 0901 0000 0037 0000 0a01 0000 003e 0000
00001d0 0b01 0000 0043 0000 0b01 0000 004e 0000
00001e0 0501 0000 0053 0000 0c02 0000 2e00 7973
00001f0 746d 6261 2e00 7473 7472 6261 2e00 6873
0000200 7473 7472 6261 2e00 6572 2e6c 6574 7478
0000210 2e00 6164 6174 2e00 7362 0073 722e 646f
0000220 7461 0061 6e2e 746f 2e65 4e47 2d55 7473
0000230 6361 006b 632e 6d6f 656d 746e 0000 0000
0000240 0000 0000 0000 0000 0000 0000 0000 0000
*
0000260 0000 0000 0000 0000 001f 0000 0001 0000
0000270 0006 0000 0000 0000 0034 0000 0059 0000
0000280 0000 0000 0000 0000 0001 0000 0000 0000
0000290 001b 0000 0009 0000 0040 0000 0000 0000
00002a0 01ac 0000 0040 0000 0009 0000 0001 0000
00002b0 0004 0000 0008 0000 0025 0000 0001 0000
00002c0 0003 0000 0000 0000 008d 0000 0000 0000
00002d0 0000 0000 0000 0000 0001 0000 0000 0000
00002e0 002b 0000 0008 0000 0003 0000 0000 0000
00002f0 008d 0000 0000 0000 0000 0000 0000 0000
0000300 0001 0000 0000 0000 0030 0000 0001 0000
0000310 0002 0000 0000 0000 008d 0000 0010 0000
0000320 0000 0000 0000 0000 0001 0000 0000 0000
0000330 0038 0000 0001 0000 0000 0000 0000 0000
0000340 009d 0000 0000 0000 0000 0000 0000 0000
0000350 0001 0000 0000 0000 0048 0000 0001 0000
0000360 0030 0000 0000 0000 009d 0000 0023 0000
0000370 0000 0000 0000 0000 0001 0000 0001 0000
0000380 0011 0000 0003 0000 0000 0000 0000 0000
0000390 01ec 0000 0051 0000 0000 0000 0000 0000
00003a0 0001 0000 0000 0000 0001 0000 0002 0000
00003b0 0000 0000 0000 0000 00c0 0000 00d0 0000
00003c0 000a 0000 0008 0000 0004 0000 0010 0000
00003d0 0009 0000 0003 0000 0000 0000 0000 0000
00003e0 0190 0000 0019 0000 0000 0000 0000 0000
00003f0 0001 0000 0000 0000
00003f8
sys32200185@embedded:~$ whoami
sys32200185
sys32200185@embedded:~$ date
2021. 10. 04. (월) 04:04:04 KST
sys32200185@embedded:~$
```

## 16. Whoami & date