

Report 2



학 번	32202546
성 명	안지성
담당교수	최종무 교수님
제 출 일	2021.10.08

- 요구 사항 1: vi 편집기로 자신이 좋아하는 시를 파일로 생성 및 내용 표시.
- 요구 사항 2:
 - 10개 이상 리눅스 명령어 실습 포함 ("ls -l, ps, pipe 등")
 - 번역 실습 포함 ("gcc, as 등")
 - 자신의 학번과 날짜 출력 ("whoami, date" 사용)

1. `vi poem.txt` -> `vi` command로 `txt`파일을 만들어서 편집.
2. `ls` -> `directory` 내부 파일들을 보여줌.
3. `cat poem.txt` -> `poem.txt`파일 내용을 보여줌.

4. `ls -l` -> directory 내부 파일들을 상세히 보여줌.
5. `ps` -> 현재 사용하는 process 목록을 보여줌.

```

sys32202546@embedded:~$ mkdir test
sys32202546@embedded:~$ ls
examples.desktop  poem.txt  test
sys32202546@embedded:~$ cp poem.txt test
sys32202546@embedded:~$ ls
examples.desktop  poem.txt  test
sys32202546@embedded:~$ cd test
sys32202546@embedded:~/test$ ls
poem.txt

```

6. mkdir test -> test directory 생성.

7. cp poem.txt test -> test directory에 poem.txt파일 복사.

8. cd test -> test directory로 이동.

```

sys32202546@embedded:~/test$ cd ..
sys32202546@embedded:~$ ls
examples.desktop  poem.txt  test
sys32202546@embedded:~$ rmdir test
rmdir: failed to remove 'test': Directory not empty
sys32202546@embedded:~$ cd test
sys32202546@embedded:~/test$ ls
poem.txt
sys32202546@embedded:~/test$ rm poem.txt
sys32202546@embedded:~/test$ ls
sys32202546@embedded:~/test$ cd ..
sys32202546@embedded:~$ ls
examples.desktop  poem.txt  test
sys32202546@embedded:~$ rmdir test
sys32202546@embedded:~$ ls
examples.desktop  poem.txt

```

9. cd .. -> 상위 directory로 이동.

10. rm dir test -> directory 내부에 파일이 존재하면 directory 삭제 불가
-> cd test -> rm poem.txt로 파일 삭제 후 다시 시도 -> 성공!

```

sys32202546@embedded:~$ grep -n 별 poem.txt | wc -l
2
sys32202546@embedded:~$ grep -n 별 poem.txt
6:그 토록 부르던 별과 꿈, 그런 것들..
7:별은 과녁이었고 꿈은 그저 꿈이라 부르기에 알맞은 거였다고 말해줄래.

```

11. grep -n 별 poem.txt | wc -l

-> "별" 키워드를 grep command를 통해 poem.txt에서 찾은 후, pipe로 다음 command인 wc -l에 넘겨주며, wc -l은 "별" 키워드가 들어있는 줄의 개수를 출력해준다.

12. grep -n 별 poem.txt

-> "별" 키워드를 grep command를 통해 poem.txt에서 찾은 후, 줄 번호와 내용 출력.

```

sys32202546@embedded:~$ vi test.c
sys32202546@embedded:~$ ls
examples.desktop poem.txt test.c
sys32202546@embedded:~$ cat test.c
#include <stdio.h>

int a,b,c;

main(){
    a = 20000426;
    b = 20011029;
    c = 20210505;
    printf("U:%d O:%d D:%d\n",b,c,a);
}

```

13. vi test.c -> vi 편집기를 사용하여 c프로그램 작성.

14. cat test.c

```

sys32202546@embedded:~$ gcc -S test.c
sys32202546@embedded:~$ ls
examples.desktop poem.txt test.c test.s
sys32202546@embedded:~$ more test.s
.file "test.c"
.section .rodata
.LC0:
.string "U:%d O:%d D:%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     $20000426, a
    movl     $20011029, b
    movl     $20210505, c
    movl     a, %eax
    movl     %eax, 12(%esp)
    movl     c, %eax
    movl     %eax, 8(%esp)
    movl     b, %eax
    movl     %eax, 4(%esp)
    movl     $.LC0, (%esp)
    call     print
    leave
    ret
    .size    main, .-main
    .comm    a,4,4
    .comm    b,4,4
    .comm    c,4,4
    .section .note.GNU-stack,"",@progbits
    .ident   "GCC: (GNU) 3.4.6 (Debian 3.4.6-5)"

```

15. gcc -S test.c -> test.s라는 assembly 파일을 생성.

16. more test.s

```

sys32202546@embedded:~$ gcc test.c
sys32202546@embedded:~$ ls
a.out  examples.desktop  poem.txt  test.c  test.o  test.s
sys32202546@embedded:~$ ./a.out
U:20011029 O:20210505 D:20000426

```

17. gcc test.c -> 실행가능한 a.out파일을 gcc로 컴파일하여 생성.

18. ./a.out -> 현재 directory에 있는 a.out 실행.

```

sys32202546@embedded:~$ gcc -o test.out test.c
sys32202546@embedded:~$ ls
a.out  examples.desktop  poem.txt  test.c  test.o  test.out  test.s
sys32202546@embedded:~$ ./test.out
U:20011029 O:20210505 D:20000426

```

19. gcc -o test.out test.c -> test.c를 gcc -o로 컴파일하여 실행가능한 test.out을 생성.

20. ./test.out

```

sys32202546@embedded:~$ ls
a.out  examples.desktop  poem.txt  test.c  test.out  test.s
sys32202546@embedded:~$ as -o test.o test.s
sys32202546@embedded:~$ ls
a.out  examples.desktop  poem.txt  test.c  test.o  test.out  test.s
sys32202546@embedded:~$ more test.o

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

```

21. as -o test.o test.s -> assembly 파일 test.s로 binary 파일 test.o 생성.

```

sys32202546@embedded:~$ hexdump test.o
00000000 457f 464c 0101 0001 0000 0000 0000 0000
00000010 0001 0003 0001 0000 0000 0000 0000 0000
00000020 0248 0000 0000 0000 0034 0000 0000 0028
00000030 000b 0008 8955 83e5 18ec e483 b8f0 0000
00000040 0000 c083 830f 0fc0 e8c1 c104 04e0 c429
00000050 05c7 0000 0000 2eaa 0131 05c7 0000 0000
00000060 5815 0131 05c7 0000 0000 6349 0134 00a1
00000070 0000 8900 2444 a10c 0000 0000 4489 0824
00000080 00a1 0000 8900 2444 c704 2404 0000 0000
00000090 fce8 ffff c9ff 55c3 253a 2064 3a4f 6425
000000a0 4420 253a 0a64 0000 4347 3a43 2820 4e47
000000b0 2955 3320 342e 362e 2820 6544 6962 6e61
000000c0 3320 342e 362e 352d 0029 0000 0000 0000
000000d0 0000 0000 0000 0000 0000 0000 0001 0000
000000e0 0000 0000 0000 0000 0004 fff1 0000 0000
000000f0 0000 0000 0000 0000 0003 0001 0000 0000
00001000 0000 0000 0000 0000 0003 0003 0000 0000
00001010 0000 0000 0000 0000 0003 0004 0000 0000
00001020 0000 0000 0000 0000 0003 0005 0000 0000
00001030 0000 0000 0000 0000 0003 0006 0000 0000
00001040 0000 0000 0000 0000 0003 0007 0008 0000
00001050 0000 0000 0063 0000 0012 0001 000d 0000
00001060 0004 0000 0004 0000 0011 fff2 000f 0000
00001070 0004 0000 0004 0000 0011 fff2 0006 0000
00001080 0004 0000 0004 0000 0011 fff2 0011 0000
00001090 0000 0000 0000 0000 0010 0000 7400 7365
000010a0 2e74 0063 616d 6e69 6100 6200 7000 6972
000010b0 746e 0066 001e 0000 0901 0000 0028 0000
000010c0 0a01 0000 0032 0000 0b01 0000 003b 0000
000010d0 0901 0000 0044 0000 0b01 0000 004d 0000
000010e0 0a01 0000 0058 0000 0501 0000 005d 0000
000010f0 0c02 0000 2e00 7973 746d 6261 2e00 7473
00002000 7472 6261 2e00 6873 7473 7472 6261 2e00
00002010 6572 2e6c 6574 7478 2e00 6164 6174 2e00
00002020 7362 0073 722e 646f 7461 0061 6e2e 746f

```

22. hexdump test.o

-> more command는 text파일만 사용할 수 있으므로 binary파일은 hexdump command를 사용하여 읽고 출력한다.

```
sys32202546@embedded:~$ whoami
sys32202546
sys32202546@embedded:~$ date
2021. 10. 08. (금) 16:57:28 KST
```

23. whoami & date -> id와 현재 날짜&시간을 출력

감사합니다!