1번

1-1)

rdbuff f1, buffer

clear x

td = x'f1'

jeq \* -3

rd = x'f1'

stch buffer, x

1-2)

rdbuff f2, area

clear x

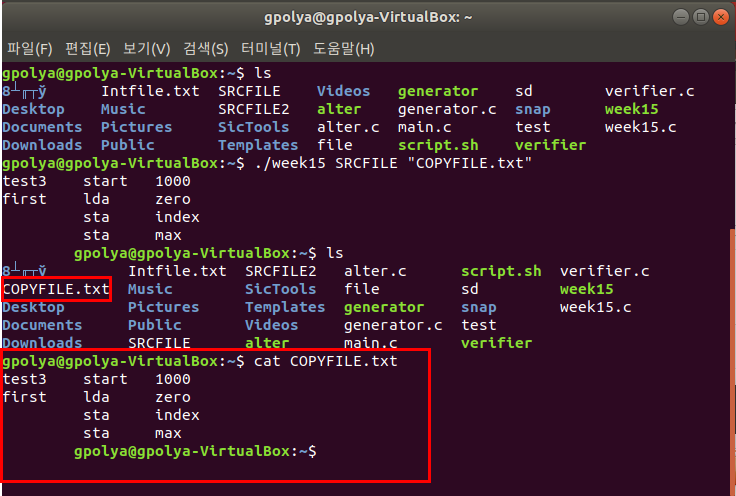
td = x'f2'

jeq \* -3

rd = x'f2'

stch area, f2

2번 실행화면



코드분석)

int main(int argc, char\* argv[])는 command line argument로 받아오기 위한 것

fd = open(argv[1],O\_RDONLY)는 argv[1]이름을 가지는 파일을 연다. 두번째 argument

open함수는 sys/file.h 헤더에 정의되어있다.

open함수의 첫번째 parameter는 가져올 파일 이름, 두번째 파라미터는 파일에 대한 열기 옵션이다. 파일 열기옵션은 O\_RDONLY는 읽기 전용, O\_CREAT는 해당하는 파일이 없으면 파일을 생성, O\_RDWR은 읽기쓰기 모두가능한 옵션이다.

open함수의 반환값은 정상으로 반환했다면 파일 디스크립터의 양의 정수 값, 실패하면 -1을 반환한다.

charsRead = read(fd,buffer,100)은 buffer에 fd의 내용을 저장한다.

첫번째 파라미터는 데이터를 전송해 주는 대상을 가리키는 파일 디스크립터…여기선 fd이다.

파일 디스크립터가 뭔지 잘 이해 못하겠는데

윈도우에서 FILE \*fp = fopen(“test.txt”,”w”)에서 fp의 역할과 비슷하다고 이해했다.

두번째 파라미터는 수신한 데이터를 저장할 버퍼를 가리키는 포인터.

세번째 파라미터는 수신할 최대 바이트 수. 즉 파일에서 읽어올 최대 바이트 수를 의미

read함수의 리턴값은 성공시 수신한 바이트 수, 즉 이때는 100, 실패시 -1을 반환한다.

write(1,buffer,charsRead)

write함수도 read함수와 비슷하다.

다만 write함수의 첫번째 파라미터를 1로 주니까 우분투 터미널창에 출력되더라

close(fd)는 파일디스크립터를 닫는다. fclose(fp)하고 비슷

fd = open(argv[2], O\_CREAT|O\_RDWR,0600)은 argv[2]이름의 파일이 없으면 그 파일을 생성하고 읽기쓰기 모두 가능하게 한다. 이때 마지막 파라미터 0600은 접근권한을 의미한다.

write(fd,buffer,100)은 이제 buffer에 저장되어있던 것들을 100바이트만큼 fd에 출력한다.

여기서 fd는 argv[2]이름을 가진 파일이므로 여기에 buffer의 내용이 저장될 것이다.

close(fd)는 파일디스크립터 닫기

참고한 자료

<http://forum.falinux.com/zbxe/index.php?_filter=search&mid=C_LIB&search_target=title&search_keyword=open&document_srl=408448>

<https://mintnlatte.tistory.com/287>

<https://m.blog.naver.com/PostView.nhn?blogId=bestheroz&logNo=113881732&proxyReferer=https%3A%2F%2Fwww.google.com%2F>