

어셈블리어

최종과제

과목명 : 어셈블리어

담당교수 : 김기태

분반 : 02

학번 : 12080912

학과 : 컴퓨터정보공학

이름 : 남건백

제출일 : 2009.06.21

12080912 컴퓨터정보공학과 남건백

**1. 제목 : 도스에디터**

**2. 주제설명**

도스상에서 문서를 편집, 저장 및 불러오기가 가능한 문서편집기입니다.

이 주제를 선정한 이유는 한 학기 동안 어셈블리어 강의를 들으면서 주로 다룬 내용들을 살펴보니 대부분 문자를 입력 받고 출력하고 문자를 조작하는 내용들이 제일 많았기 때문에 도스상에서 문서를 편집할 수 있는 도스문서편집기를 프로그래밍 하는 것이 지금까지 수업내용을 많이 반영 할 수 있을 것 같아 문서편집기를 만들게 되었습니다.

**3. 프로그램 구성**

프로그램의 전체적인 구성은 프로그래밍을 시작하기 전에 기존 도스에디트의 기본 기능들을 구현하는데 목표를 두었습니다 또한 프로그래밍 환경은 윈도우 비스타 x86 SP1, 에디트플러스3, 그리고 컴파일은 수업시간에 주신자료를 통해 컴파일 하였습니다.

에디터의 기능들을 소개하면

1. 문자 입력기능(키보드를 통한 문자입력)
2. 방향키 기능  
   (커서이동, 커서가 텍스트영역 경계 면에서 갈 수 없는 곳으로 이동할 시 이전 줄/다음 줄로 이동)
3. 엔터 키 기능 (커서를 다음줄 첫 번째 칸으로 이동시킴)
4. 백스페이스 키 기능 (보통 문서편집기에서의 백스페이스와 동일)
5. 딜리트 키 기능 (보통 문서편집기에서의 딜리트와 동일)
6. 홈 키 기능 (에디터의 첫 줄, 첫 칸으로 커서이동)
7. 엔드 키 기능 (에디터의 마지막 줄, 마지막 칸으로 커서이동)
8. 메뉴 기능 (F1를 누르면 문서편집 도중에 화면에 메뉴창이 출력됨

8-1 메뉴창 닫기 기능 (메뉴창을 닫고 다시 문서편집으로 돌아가는 기능)

8-2 새파일 기능 (지금까지 작성한 문서를 버리고 새로운 문서를 가져오는 기능)

8-3 저장 기능 (지금까지 편집한 문서의 내용을 저장하는 기능, 파일명 지정 가능)

8-4 불러오기 기능 (저장해놓은 문서의 내용을 불러오는 기능, 파일명 지정 가능)

8-5 끝내기 기능 (에디트를 닫고 프로그램을 종료시키는 기능)

1. 종료 기능 (ESC를 누르면 프로그램을 종료시키는 기능)
2. 행,열 표시기능 (화면의 오른쪽 하단에 현재 커서의 행,열의 위치를 표시해줌)
3. 문서의 제목표시 기능   
   (화면 상단에 현재 편집 하고있는 문서의 제목을 표시해줌, 세이브, 로드시에 제목변경)

**4. 소스(With 주석)**

TITLE 12080912 FINAL

.MODEL SMALL

.STACK 64

.DATA

NEW\_FILE\_TITLE DB ' UNTITLED TEXT FILE ' ;새파일 제목 텍스트

FILE\_TITLE DB 22 DUP (' ') ;파일 제목 텍스트

FRAME DB 0C9H, 2 DUP ( 0CDH ), ' EDITOR : ', 66 DUP ( 0CDH ), 0BBH ;에디터 틀

DB 22 DUP( 0BAH, 78 DUP (' '), 0BAH )

MENU\_STR DB ' ', 0AFH,' Press F1-Menu', 46 DUP(' '),'|ROW: COL: '

HELP\_MENU\_STR DB ? ;도움말 텍스트

COUNT\_ROW DB ?

ROW DB 1 ;현재 행

COL DB 1 ;현재 열

LEFT\_LIM EQU 1 ;왼쪽 한계

RIGHT\_LIM EQU 4EH ;오른쪽 한계

TOP\_LIM EQU 1 ;위쪽 한계

BTM\_LIM EQU 22 ;아래쪽 한계

ROWLEN DB 78 ;한줄의 칸 갯수

ESC\_CHK DB 0 ;ESC가 눌렸는지 체크하는 변수

TEXT\_AREA DB 1800 DUP (' '), 0DH, 0AH ; 텍스트 저장영역

CALROW DB 2 DUP (' ')

CALCOL DB 2 DUP (' ')

;------------------------------------------------

MTOPROW EQU 08 ;메뉴에서의 윗줄

MBOTROW EQU 14 ;메뉴에서의 아랫줄

MLEFCOL EQU 30 ;화면끝에서 떨어진 칸수

MATTRIB DB ? ;메뉴 속성을 받을 변수

MROW DB 00 ;메뉴에서 사용할 행

MESC\_CHK DB 0

SHADOW DB 19 DUP (0B1H ) ;메뉴의 그림자

MENU DB 0C9H, 17 DUP ( 0CDH ), 0BBH ;메뉴

DB 0BAH, ' Continue ', 0BAH

DB 0BAH, ' New file ', 0BAH

DB 0BAH, ' Save file ', 0BAH

DB 0BAH, ' Load file ', 0BAH

DB 0BAH, ' Exit editor ', 0BAH

DB 0C8H, 17 DUP ( 0CDH ), 0BCH

;--------------------------------------------------

SAVEMESS DB 'Save the data in C:\','$' ;세이브 시에 출력될 메시지

LOADMESS DB 'Load the data from C:\','$' ;로드 시에 출력될 메시지

FILEMESS DB 'Enter the file name for Save/Load','$' ;세이브,로드시 파일이름 입력안내 메시지

ENDMESS DB '<Press any key> Program exiting now...' ;프로그램 종료시 출력될 메시지

PRESS DB '<Press any key>','$' ;키 입력 메시지

ERRORMESS DB 'File iperation error! ','$' ;파일 연산중 에러가 발생될 때

FILEPATH DB 'C:\'

INPUTPATH DB 23 DUP(' '), 00H ;아스키제로 스트링

FILEHAND DW ? ;파일핸들 저장변수

PARLIST LABEL BYTE

MAX\_LEN DB 20

ACT\_LEN DB ?

KB\_DATA DB 21 DUP(' ')

.386 ;-------------------------------------------

.CODE

A10MAIN PROC FAR

MOV AX,@DATA

MOV DS,AX

MOV ES,AX

MOV AH,00H ;비디오 모드 설정

MOV AL,03H

INT 10H

MOV AH,05H ;활성 페이지 선택

MOV AL,00H

INT 10H

A20:

CALL B10ALLCLEAR ;전체 디스플레이 초기화

CALL D10FRAME ;에디터 틀 출력(제목,단축키 설명 포함)

A30:

MOV DH,ROW ;커서 설정

MOV DL,COL

CALL E10CURSOR

CALL G10KEYBRD ;입력받기

;입력받은거 메모리에 넣고

;메모리의 내용 출력

CALL I10DISCHR

CMP ESC\_CHK,1 ;ESC가 눌렸는지 확인한다(눌리면 종료)

JE A90

JMP A30

MOV AH,10H

INT 16H

A90:

MOV AX,1301H

MOV BH,00

MOV BL,0F4H

LEA BP,ENDMESS

MOV CX,38

MOV DX,0A14H

INT 10H

;종료 메시지를 출력후

MOV AH,10H ;아무키나 누르면 스크롤후 종료

INT 16H

CALL B10ALLCLEAR

MOV AX, 4C00H

INT 21H

A10MAIN ENDP

;------------------------------------------------

B10ALLCLEAR PROC NEAR

;프로시져를 호출하면 화면 전체를 스크롤

;화면전체를 스크롤(텍스트 영역 밖까지)

PUSHA

MOV AX,0600H

MOV BH,4FH

MOV CX,0000H

MOV DX,184FH

INT 10H

POPA

RET

B10ALLCLEAR ENDP

;------------------------------------------------

C10CLEAR PROC NEAR

;AL를 세팅하면 호출하면 AL값만큼 화면 스크롤

;텍스트 영역안에서만 스크롤

PUSHA

MOV AH,06H

MOV BH,4FH

MOV CX,0101H

MOV DX,174EH

INT 10H

POPA

RET

C10CLEAR ENDP

;------------------------------------------------

D10FRAME PROC NEAR

;에디터의 전체틀을 출력

;커서 설정

PUSHA

MOV DX,0000

CALL E10CURSOR

;에디터의 바깥쪽 틀을 출력

MOV AX,1301H

MOV BL,4FH

LEA BP,FRAME

MOV CX,1920

MOV DX,0000H

INT 10H

;밑줄메뉴단축키 출력

MOV AX,1301H

MOV BL,0F4H

LEA BP,MENU\_STR

MOV CX,80

MOV DX,1700H

INT 10H

;문서 제목(타이틀) 출력

MOV AX,1301H

MOV BL,0F4H

LEA BP,NEW\_FILE\_TITLE

MOV CX,20

MOV DX,000CH

INT 10H

POPA

RET

D10FRAME ENDP

;------------------------------------------------

E10CURSOR PROC NEAR

;DX를 세팅하고 호출하면 커서의 위치를 설정

PUSHA

MOV AH,02

MOV BH,00

INT 10H

POPA

RET

E10CURSOR ENDP

;------------------------------------------------

F10INPUTCHR PROC NEAR

;텍스트영역에 표시할 데이터세그먼트 TEXT\_AREA의 영역에 값 입력

PUSHA

PUSH AX ;곱하기에서 AX를 사용하므로 백업

MOV AX,0000 ;AX 초기화

MOVZX AX,ROW ;AX = (ROW-1)\*78

DEC AX

IMUL ROWLEN

MOVZX BX,COL ;AX = AX + (COL-1)

ADD AX,BX

DEC AX

MOV CX,AX

POP AX ;백업해놨던 AX를 복원

LEA DI,TEXT\_AREA ;STOSB를 통해서 TEXT\_AREA에 삽입

ADD DI,CX

CLD

STOSB

;COL값을 계산해서 COL이 79까지만 계산되고

;79를 넘어가면 ROW를 증가시켜 행을 나타내게한다.

CMP COL,RIGHT\_LIM ;마지막 열인지 검사해서 마지막열이면 다음행으로

JB F10

CMP ROW,BTM\_LIM ;마지막줄, 마지막열에선 제자리에 멈추기위해서 검사

JE F90 ;(증가없이 바로리턴)

INC ROW

MOV COL,01

JMP F90

F10:

INC COL

F90:

POPA

RET

F10INPUTCHR ENDP

;------------------------------------------------

G10KEYBRD PROC NEAR

;키보드로부터 입력을받아들여서 그에 맞는 기능을 수행하는

;프로시져를 호출한다

PUSHA

MOV AH,10H

INT 16H ;입력을 받아들인다

CMP AL,00H ;받아들인 문자가 알파벳인지 검사하는 과정

JE G20

CMP AL,0E0H

JE G20

CMP AL,0DH

JE G20

CMP AL,08H

JE G40

CMP AL,1BH

JE G70

CALL F10INPUTCHR ;알파벳일때는 텍스트영역에 문자삽입

JMP G90

G20: ;오른쪽 화살표

CMP AH,4DH

JNE G30

CALL H10RTARRW

JMP G90

G30: ;왼쪽 화살표

CMP AH,4BH

JNE G33

CALL H20LFARRW

JMP G90

G33: ;아래쪽 화살표

CMP AH,50H

JNE G36

CALL H40DWNARRW

JMP G90

G36: ;위쪽 화살표

CMP AH,48H

JNE G39

CALL H30UPARRW

JMP G90

G39: ;엔터

CMP AH,1CH

JNE G40

CALL H80ENTER

JMP G90

G40: ;백스페이스

CMP AH,0EH

JNE G45

CALL H90BACKSP

JMP G90

G45: ;딜리트

CMP AH,53H

JNE G50

CALL H70DELETE

JMP G90

G50: ;홈

CMP AH,47H

JNE G60

CALL H100HOME

JMP G90

G60: ;엔드

CMP AH,4FH

JNE G65

CALL H110END

MOV ROW,24

JMP G90

G65: ;F1-메뉴

CMP AH,3BH

JNE G70

CALL CAMENU

JMP G90

G70: ;ESC-종료

CMP AH,01H

JNE G90

MOV ESC\_CHK,1

G90:

POPA

RET

G10KEYBRD ENDP

;------------------------------------------------

H10RTARRW PROC NEAR

;오른쪽 화살표에 대한 동작을 수행함

PUSHA

CMP COL, RIGHT\_LIM ;오른쪽 끝인가에 대한 검사

JAE H11 ;끝이면 아랫줄 첫칸으로 커서이동

INC COL

JMP H19

H11:

CMP ROW,BTM\_LIM ;맨마지막줄은 예외처리

JE H19 ;바로 리턴

CALL H40DWNARRW

CALL H50FLINE

H19:

POPA

RET

H10RTARRW ENDP

;------------------------------------------------

H20LFARRW PROC NEAR

;왼쪽 화살표에 대한 동작을 수행함

PUSHA

CMP COL, LEFT\_LIM ;왼쪽 끝인가에 대한 검사

JBE H21 ;끝이면 윗줄 마지막칸으로 커서이동

DEC COL

JMP H29

H21:

CMP ROW,TOP\_LIM ;맨윗줄에선 예외처리

JE H29 ;바로 리턴

CALL H30UPARRW

CALL H60ELINE

H29:

POPA

RET

H20LFARRW ENDP

;------------------------------------------------

H30UPARRW PROC NEAR

;위쪽 화살표에 대한 동작을 수행함

CMP ROW, TOP\_LIM

JE H39

H31:

DEC ROW

H39:

RET

H30UPARRW ENDP

;------------------------------------------------

H40DWNARRW PROC NEAR

;아래쪽 화살표에 대한 동작을 수행함

CMP ROW, BTM\_LIM

JE H49

H41:

INC ROW

H49:

RET

H40DWNARRW ENDP

;------------------------------------------------

H50FLINE PROC NEAR

;줄의 맨앞으로 커서 이동

MOV COL, LEFT\_LIM

CALL E10CURSOR

RET

H50FLINE ENDP

;------------------------------------------------

H60ELINE PROC NEAR

;줄의 맨뒤로 커서 이동

MOV COL, RIGHT\_LIM

CALL E10CURSOR

RET

H60ELINE ENDP

;------------------------------------------------

H70DELETE PROC NEAR

;커서 뒤에 있는 한문자를 지우는 기능

PUSHA

MOV AX,0000 ;AX 초기화

MOVZX AX,ROW ;AX = (ROW-1)\*78

DEC AX

IMUL AX,78

MOVZX BX,COL ;AX = AX + (COL-1)

ADD AX,BX

DEC AX

MOV BX,AX

MOV CX,1794

SUB CX,AX

LEA DI, [TEXT\_AREA+BX] ;한문자씩 반복해서 끌어옴

LEA SI, [TEXT\_AREA+BX+1]

REP MOVSB

POPA

RET

H70DELETE ENDP

;------------------------------------------------

H80ENTER PROC NEAR

;에디터의 엔터기능

PUSHA

CMP ROW,BTM\_LIM ;마지막줄에선 넘어가지않게 예외처리

JE H89 ;마지막줄이면 넘어가기

MOV COL,1

INC ROW

H89:

POPA

RET

H80ENTER ENDP

;------------------------------------------------

H90BACKSP PROC NEAR

;백스페이스의 기능

;백스페이스를 COL를 감소시키고 딜리트를 호출하므로써 구현

CMP ROW,TOP\_LIM ;맨 첫이고

JNE H91

CMP COL,LEFT\_LIM ;맨 첫칸이면 예외처리

JE H99

H91:

JE H99

CALL H20LFARRW

CALL H70DELETE ;딜리트 호출

H99:

RET

H90BACKSP ENDP

;------------------------------------------------

H100HOME PROC NEAR\

;홈키의 기능을 한다

;텍스트영역의 첫줄,첫칸으로 커서 이동

MOV ROW,1

MOV COL,1

RET

H100HOME ENDP

;------------------------------------------------

H110END PROC NEAR

;엔드키의 기능을 한다.

;텍스트영역의 마지막줄,마지막열로 커서 이동

MOV ROW,22

MOV COL,77

RET

H110END ENDP

;------------------------------------------------

I10DISCHR PROC NEAR

;TEXT\_AREA에 있는 모든 문자를 출력한다.

;더불어 하단의 메뉴단축키 줄의 ROW,COL을 출력한다.

PUSHA

MOV DX,0101H

CALL E10CURSOR

;첫번째줄 출력

MOV AX,1301H

MOV BL,4FH

LEA BP,TEXT\_AREA

MOV CX,78

MOV DX,0101H

PUSHA

INT 10H

POPA

MOV COUNT\_ROW,21

;두번째 줄부터는

;루프를 통해 반복해서 행을 계산, 출력

I20:

ADD BP,78

ADD DX,0100H

PUSHA

INT 10H

POPA

CMP COUNT\_ROW,0

DEC COUNT\_ROW

JNE I20

CALL J10ROWCOL

POPA

RET

I10DISCHR ENDP

;------------------------------------------------

J10ROWCOL PROC NEAR

PUSHA

MOV AL,ROW ;ROW 계산을 위해 AL에 ROW를 대입

MOV CL,0

CMP AL,0AH ;ROW > 10인가 검사

JB J20

J15:

SUB AL,0AH ;ROW가 10이상일때

INC CL ;AL에 10을 빼고(AL은 1자리를 나타냄)

CMP AL,0AH ;CL를 증가 (CL는 2자리를 나타냄)

JAE J15

J20:

ADD AL,30H

MOV CALROW+1,AL ;CALROW+1는 1자리수

ADD CL,30H ;CALROW는 2자리수

MOV CALROW,CL

MOV AL,COL ;COL 계산을 위해 AL에 ROW대입

MOV CL,0 ;계산방식은 ROW와 동일

CMP AL,0AH

JB J30

J25:

SUB AL,0AH

INC CL

CMP AL,0AH

JAE J25

J30:

ADD AL,30H

MOV CALCOL+1,AL

ADD CL,30H

MOV CALCOL,CL

MOV AX,1301H ;CALROW,CALCOL 출력

MOV BX,00F4H

LEA BP,CALROW

MOV CX,2

MOV DH,23

MOV DL,68

PUSHA ;레지스터세팅을 재활용하기위한

INT 10H ;PUSHA,POPA

POPA

LEA BP,CALCOL ;BP에 COL주소 입력

MOV DL,76

INT 10H ;COL의 출력

POPA

RET

J10ROWCOL ENDP

;-------------------------------------------------

CAMENU PROC NEAR

PUSHA

CA20: CALL M10MENU ;메뉴화면을 화면에 출력하는 프로시저 호출

MOV MROW, MTOPROW+1 ;메뉴의 처음 항목을 가리키도록 설정

MOV MATTRIB, 40H ;속성을 설정하여 해당 항목의 색 반전

CALL MDISPLY

CALL M10INPUT ;각 키입력을 받을 프로시저 호출

CMP MESC\_CHK, 1 ;메뉴 종료 체크

JNE CA20

POPA

CALL I10DISCHR ;'ESC'키로 메뉴를 종료시켰을 경우

MOV ROW, 1 ;첫줄 첫칸에 커서 설정

MOV COL, 1

CALL E10CURSOR

RET

CAMENU ENDP

;--------------------------------------------------------------------

M10MENU PROC NEAR

;메뉴 화면 출력 프로시저

PUSHA

MOV AX, 1301H ;그림자 출력

MOV BX, 0040H

LEA BP, SHADOW

MOV CX, 19

MOV DH, MTOPROW+1 ;메뉴의 한줄 아래

MOV DL, MLEFCOL+1 ;오른쪽으로 한칸 옆에 출력

MM20:

PUSHA

INT 10H

POPA

INC DH

CMP DH, MBOTROW+2 ;메뉴의 아래줄까지 그림자가 출력됐는지 확인

JNE MM20

MOV MATTRIB, 0F0H ;메뉴의 출력

MOV AX, 1300H

MOVZX BX, MATTRIB

LEA BP, MENU

MOV CX, 19

MOV DH, MTOPROW ;메뉴의 가장 윗줄

MOV DL, MLEFCOL ;가장 왼쪽 칸

MM30: PUSHA ;반복해서 메뉴끝까지 출력

INT 10H

POPA

ADD BP, 19

INC DH

CMP DH, MBOTROW+1 ;메뉴가 다 출력됐는지 확인

JNE MM30

POPA

RET

M10MENU ENDP

;--------------------------------------------------------------------

M10INPUT PROC NEAR

;메뉴에서 각 키를 입력받아 동작하는 프로시저

MC20: MOV AH, 10H

INT 16H ;키를 입력받음

CMP AH, 50H ;아래로 내려가는 방향키인지 확인

JE MC30

CMP AH, 48H ;위로 올라가는 방향키인지 확인

JE MC40

CMP AL, 0DH ;'ENTER'키인지 확인

JE MC90

CMP AL, 1BH ; ESC-종료

JE MC00

JMP MC20

MC30: MOV MATTRIB, 0F0H ;속성설정

CALL MDISPLY

INC MROW ;한줄 아래로 이동

CMP MROW, MBOTROW-1 ;메뉴 가장 아래일 경우

JBE MC50

MOV MROW, MTOPROW+1 ;메뉴의 위로 올라가도록 설정

JMP MC50

MC40: MOV MATTRIB, 0F0H ;속성설정

CALL MDISPLY

DEC MROW ;한줄 위로 가도록 설정

CMP MROW, MTOPROW+1 ;메뉴의 가장 위일 경우

JAE MC50

MOV MROW, MBOTROW-1 ;메뉴의 가장 아래로 내려가도록 설정

MC50: MOV MATTRIB, 40H ;해당 항목의 색 반전

CALL MDISPLY

JMP MC20

MC90: CMP AH, 1CH ;'ENTER'키일 경우 종료

JNE MC00

CMP MROW, MTOPROW+1 ; 1번째 메뉴 - 돌아가기

JNE MC01

JMP MC00 ; 메뉴 종료

MC01: CMP MROW, MTOPROW+2 ; 2번째 메뉴 - 새파일

JNE MC02

CALL NEWFILE ; 화면&텍스트영역 초기화

JMP MC00

MC02: CMP MROW, MTOPROW+3 ; 3번째 메뉴 - 파일저장

JNE MC03

CALL SAVEFILE ;파일저장 프로시져 호출

JMP MC00

MC03: CMP MROW, MTOPROW+4 ; 4번째 메뉴 - 파일로드

JNE MC04

CALL LOADFILE ;파일로드 프로시져 호출

JMP MC00

MC04: CMP MROW, MTOPROW+5 ; 5번째 메뉴 - 종료

JNE MC00

MOV ESC\_CHK, 01 ;프로그램을 종료시킬 변수 설정

MC00: MOV MESC\_CHK, 01 ;메뉴 화면 종료시킬 변수 설정

RET

M10INPUT ENDP

;메뉴 화면에서 해당 항목의 색이 반전되도록 하는 프로시저

;--------------------------------------------------------------------

MDISPLY PROC NEAR

PUSHA

MOVZX AX, MROW ;메뉴의 현재 줄을 받아서

SUB AX, MTOPROW ;각 항목을 정확히

IMUL AX, 19 ;가리키도록 계산

LEA SI, MENU+1

ADD SI, AX

MOV AX, 1300H

MOVZX BX, MATTRIB ;해당 항목의 색을 반전

MOV BP, SI

MOV CX, 17

MOV DH, MROW

MOV DL, MLEFCOL+1 ;첫칸부터 메뉴내부의 크기까지

INT 10H ;색을 반전 시킴

POPA

RET

MDISPLY ENDP

;--------------------------------------------------------------------

NEWFILE PROC NEAR

;텍스트 영역을 초기화 하여 새파일을 연는 기능

PUSHA

MOV AL,' ' ;텍스트영역의 전영역을

LEA DI,TEXT\_AREA ;공백문자로 초기화

MOV CX,1800

REP STOSB

;문서 제목(타이틀) 출력

MOV AX,1301H

MOV BL,0F4H

LEA BP,NEW\_FILE\_TITLE

MOV CX,20

MOV DX,000CH

INT 10H

POPA

RET

NEWFILE ENDP

;--------------------------------------------------------------------

SAVEFILE PROC NEAR

;파일을 세이브 하는 프로시져

;텍스트영역의 데이터를 텍스트 문서로 저장한다.

MOV DH, 16 ;안내 메시지출력을 위한 커서설정

MOV DL, 22

CALL E10CURSOR

MOV AH, 09

LEA DX, FILEMESS ;안내 메시지

INT 21H

MOV DH, 17 ;메시지 출력을 위한 커서 설정

MOV DL, 17

CALL E10CURSOR

MOV AH, 09

LEA DX, SAVEMESS ;파일이 성공적으로 저장되었을 때

INT 21H ;화면에 성공 메시지를 출력

CALL PATH\_INPUT ;파일명을 입력받는 프로시져 호출

CALL FILECREATE ;파일 생성

JC SERROR ;에러 발생시

MOV AH, 40H ;파일에 레코드 기록

MOV BX, FILEHAND ;파일 핸들 설정

MOV CX, 1794 ;레코드의 전체길이

LEA DX, TEXT\_AREA ;저장할 레코드

INT 21H

JC SERROR

CALL FILECLOSE ;파일 닫기

MOV DH, 18 ;아무키나 누르도록 하는 메시지 출력

MOV DL, 33

CALL E10CURSOR

MOV AH, 09

LEA DX, PRESS

INT 21H

MOV AH, 10H ;키 입력

INT 16H

RET

SERROR: CALL ERROR ;에러발생시 에러 프로시처 호출

RET

SAVEFILE ENDP

;--------------------------------------------------------------------

LOADFILE PROC NEAR

;저장된 작업을 불러오는 프로시저

;저장되있는 텍스트문서를 읽어들여서

데이터영역에서 저장하고 출력한다.

MOV DH, 16 ;안내 메시지출력을 위한 커서설정

MOV DL, 22

CALL E10CURSOR

MOV AH, 09

LEA DX, FILEMESS ;안내 메시지

INT 21H

MOV DH, 17 ;파일이 성공적으로 로드되었을 때

MOV DL, 17 ;화면에 성공 메시지를 출력

CALL E10CURSOR

MOV AH, 09

LEA DX, LOADMESS

INT 21H

CALL PATH\_INPUT ;파일명을 입력받는 프로시져 호출

CALL FILEOPEN ;파일 열기

JC LERROR ;에러 발생시

MOV AH, 3FH ;파일 읽기

MOV BX, FILEHAND ;파일 핸들 설정

MOV CX, 1800 ;레코드의 전체 길이

LEA DX, TEXT\_AREA ;레코드가 읽혀질 문자열

INT 21H

JC LERROR

CALL FILECLOSE ;파일 닫기

MOV DH, 18 ;아무키나 누르도록 하는 메시지 출력

MOV DL, 33

CALL E10CURSOR

MOV AH, 09

LEA DX, PRESS

INT 21H

MOV AH, 10H ;키 입력

INT 16H

RET

LERROR: CALL ERROR ;에러발생시 에러 프로시저 호출

RET

LOADFILE ENDP

;--------------------------------------------------------------------

FILECREATE PROC NEAR

;파일을 생성하여 파일핸들을 저장하는 프로시저

MOV AH, 3CH

MOV CX, 00

LEA DX, FILEPATH ;파일의 경로 설정

INT 21H

MOV FILEHAND, AX ;생성된 파일의 파일핸들 저장

RET

FILECREATE ENDP

;--------------------------------------------------------------------

FILECLOSE PROC NEAR

;열린 파일을 닫는 프로시저

MOV AH, 3EH

MOV BX, FILEHAND ;파일 핸들 설정

INT 21H

JC CLERROR

RET

CLERROR: CALL ERROR ;에러발생시 에러 프로시저 호출

RET

FILECLOSE ENDP

;--------------------------------------------------------------------

FILEOPEN PROC NEAR

;파일을 열어서 파일핸들을 저장하는 프로시저

MOV AH, 3DH

MOV AL, 02 ;읽기/쓰기가 되도록 설정

LEA DX, FILEPATH ;파일 경로 설정

INT 21H

MOV FILEHAND, AX ;파일핸들 저장

RET

FILEOPEN ENDP

;--------------------------------------------------------------------

PATH\_INPUT PROC NEAR

PUSHA

MOV DH,17 ;흰박스를 출력하기 위한 커서설정

MOV DL,40

CALL E10CURSOR

MOV AH,09H ;흰박스 출력(AH 09, INT10)

MOV AL,' '

MOV BH,00

MOV BL,0F4H

MOV CX,20

INT 10H

MOV DH,17 ;파일명 입력을 받기위한 커서 설정

MOV DL,40

CALL E10CURSOR

MOV AH,0AH ;파일명을 입력받음

LEA DX,PARLIST ;파라리스트를 통해서

INT 21H

CLD

LEA SI,KB\_DATA ;MOVSB를 통해서 파일명변수에

LEA DI,INPUTPATH ;입력받은 값을 넣음

MOVZX CX,ACT\_LEN

REP MOVSB

MOV AL,'.' ;파일패스 + 확장자.txt을 집어넣기 위해

STOSB ;AL에 값을 넣고 STOSB를 씀

MOV AL,'t'

STOSB

MOV AL,'x'

STOSB

MOV AL,'t'

STOSB

MOV AL,00H

STOSB

MOV DX,000CH ;기존의 제목 삭제를 위한 커서설정

CALL E10CURSOR

MOV AX,09CDH ;기존의 제목 삭제

MOV BX,004FH

MOV CX,20

INT 10H

MOV AL,' '

LEA DI,FILE\_TITLE

MOV CX,22

REP STOSB

LEA DI,FILE\_TITLE+1 ;입력받은 문자를 제목출력을 위해 이동

LEA SI,KB\_DATA

MOVZX CX,ACT\_LEN ;이동반복은 ACT\_LEN만큼

REP MOVSB

MOV AX,1301H

MOV BX,00F4H

MOVZX CX,ACT\_LEN

ADD CX,2

LEA BP,FILE\_TITLE

MOV DX,000CH

PUSHA

INT 10H

POPA

POPA

RET

PATH\_INPUT ENDP

;--------------------------------------------------------------------

ERROR PROC NEAR

;에러발생시 에러메시지를 화면에 출력하는 프로시저

PUSHA

MOV AH,09

MOV DH, 6

MOV DL, 30

CALL E10CURSOR

MOV AH, 09

LEA DX, ERRORMESS ;화면에 에러메시지를 출력

INT 21H

POPA

RET

ERROR ENDP

END A10MAIN

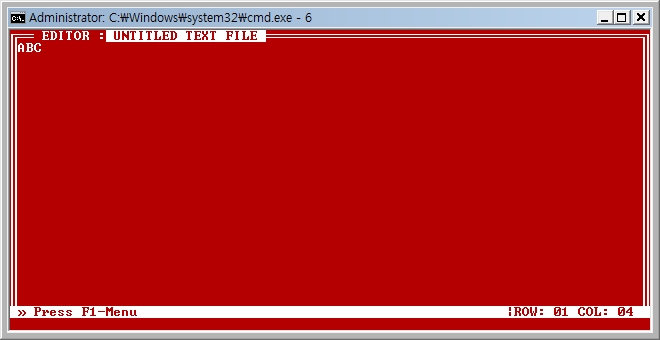
;------------------------------------------------

**5. 화면 캡쳐 및 기능 설명**



에디터 실행의 초기화면

1. 문자 입력기능(키보드를 통한 문자입력)  
   현재 초기화면에서 ABC를 입력한 상태

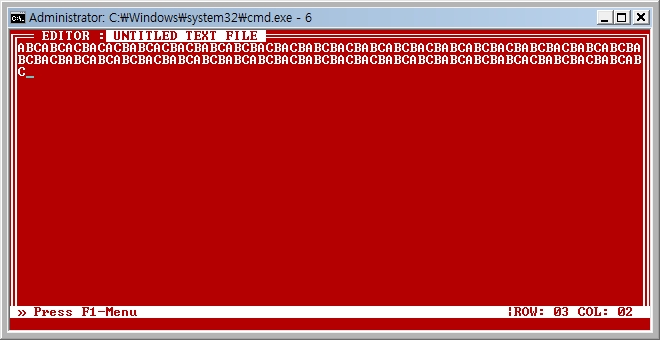


커서는 캡쳐가 되지않아 방향키 기능은 캡쳐그림을 넣지 못하였습니다.

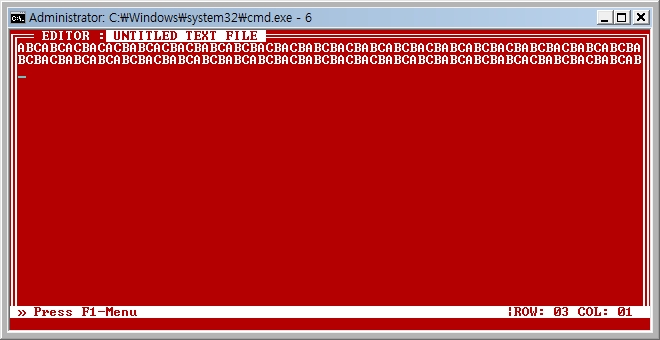
1. 방향키 기능  
   (커서이동, 커서가 텍스트영역 경계 면에서 갈 수 없는 곳으로 이동할 시 이전 줄/다음 줄로 이동)
2. 엔터 키 기능 (커서를 다음줄 첫 번째 칸으로 이동시킴)

커서는 캡쳐가 되지않아 엔터 키 기능은 캡쳐 그림을 넣지 못하였습니다.

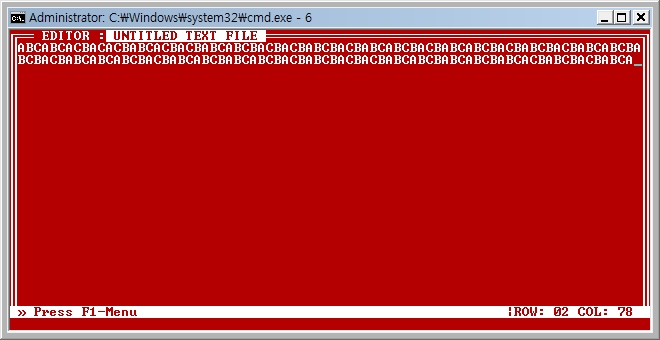
1. 백스페이스 키 기능 (보통 문서편집기에서의 백스페이스와 동일)



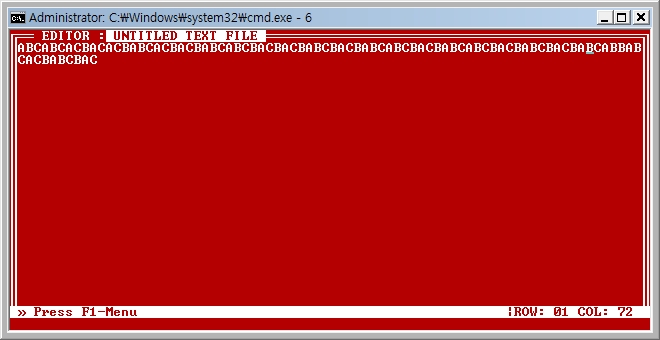
4-1 백스페이스를 눌러서 2번처럼 C가 지워지는 모습



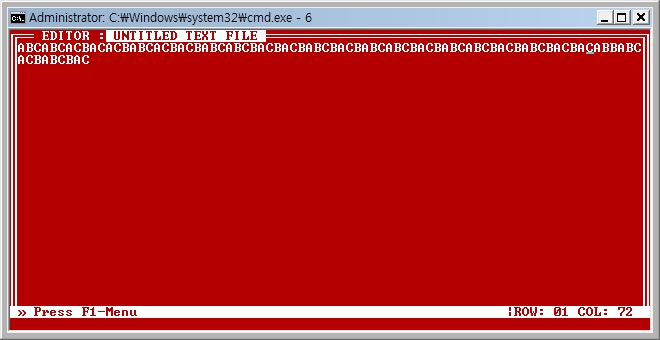
4-2 1에서 C가 지워진 상태이고 화면의 왼쪽끝에서 백스페이스를 눌러서 이전줄 끝으로 넘어가는 모습



1. 딜리트 키 기능 (보통 문서편집기에서의 딜리트와 동일)



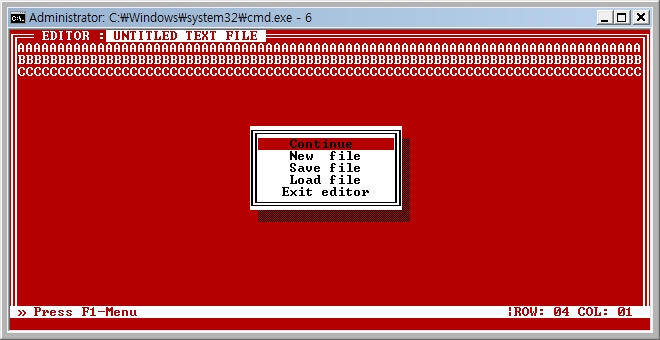
5-1 윗줄의 B에 커서를 놓고 딜리트를 눌렀을 때 B가 삭제되고 그자리를 뒤에 있던 문자들이 채우는 모습



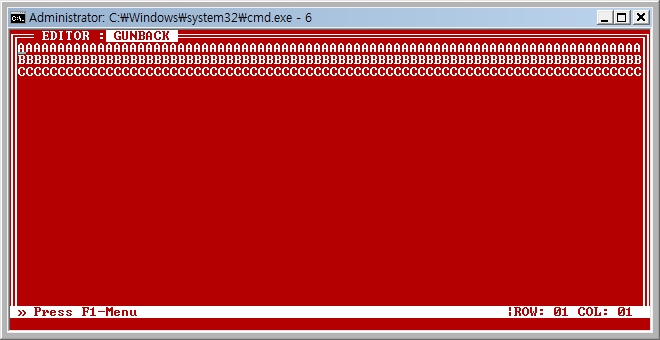
1. 홈 키 기능 (에디터의 첫 줄, 첫 칸으로 커서이동)
2. 엔드 키 기능 (에디터의 마지막 줄, 마지막 칸으로 커서이동)

홈 키와 엔드 키기능은 커서 캡쳐가 안되어 그림을 올리지 못하였습니다.

1. 메뉴 기능 (F1를 누르면 문서편집 도중에 화면에 메뉴창이 출력됨



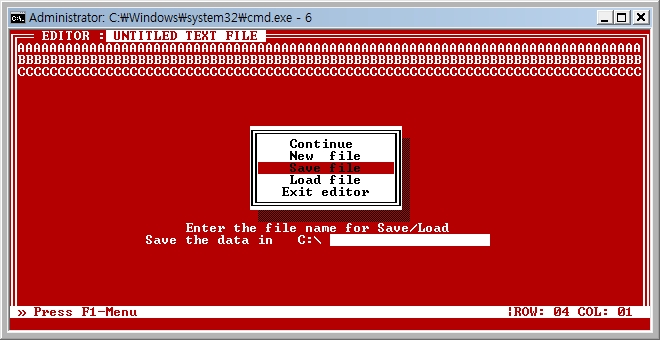
8-1 메뉴창 닫기 기능 (메뉴창을 닫고 다시 문서편집으로 돌아가는 기능)



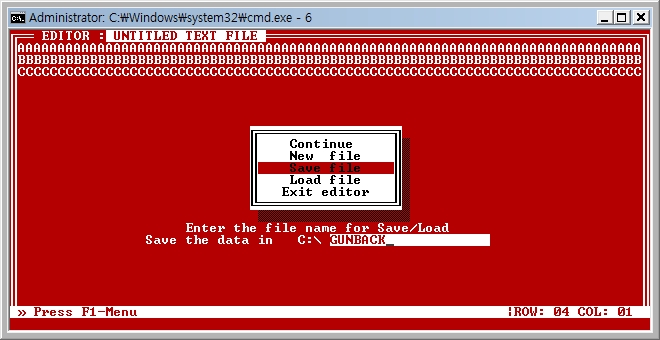
8-2 새파일 기능 (지금까지 작성한 문서의 내용을 버리고 새로운 문서를 가져오는 기능)



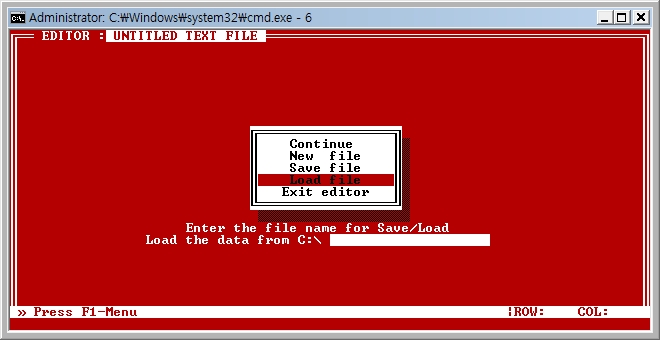
8-3 저장 기능 (지금까지 편집한 문서의 내용을 저장하는 기능)



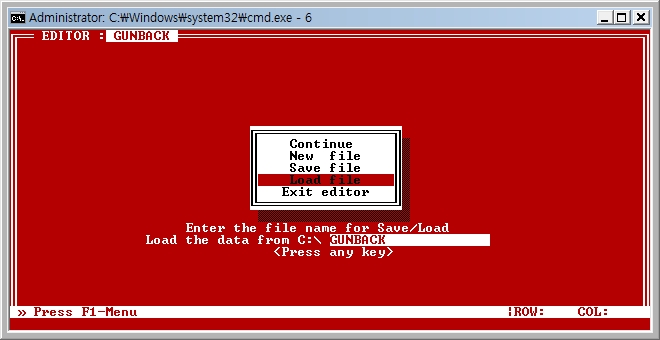
8-3-1 저장기능을 실행시킨 화면 하얀부분에 저장할 파일명을 입력받음



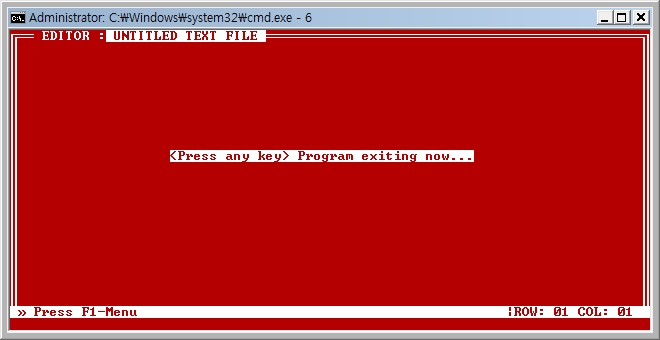
8-4 불러오기 기능 (저장해놓은 문서의 내용을 불러오는 기능)



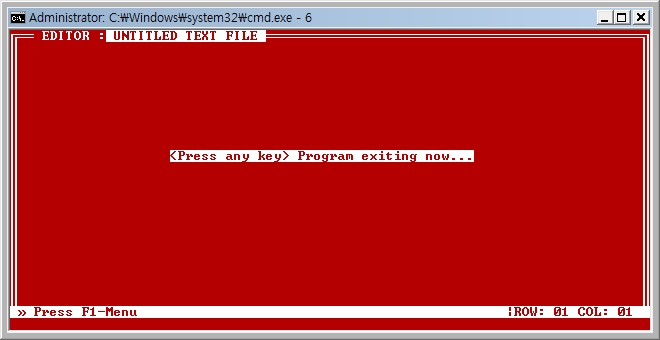
8-4-1 불러올 파일명을 입력받는 모습



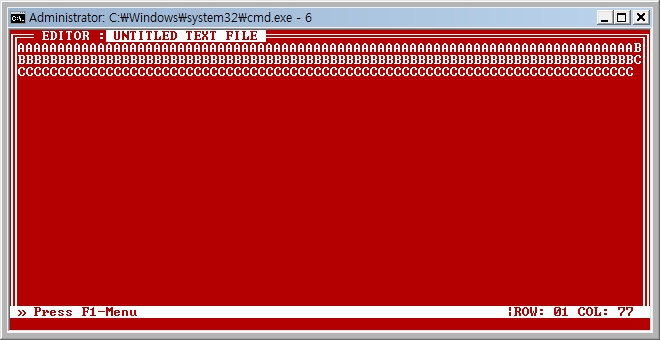
8-5 끝내기 기능 (에디트를 닫고 프로그램을 종료시키는 기능)

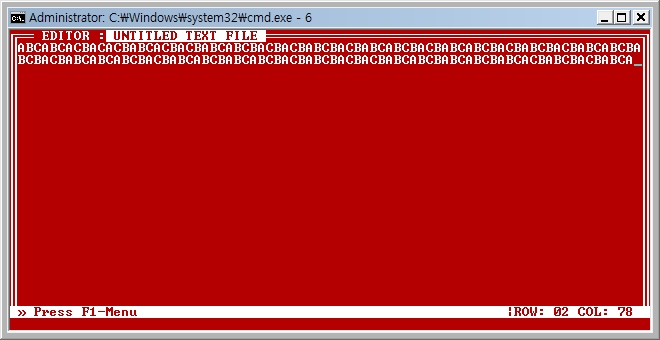


1. 종료 기능 (ESC를 누르면 프로그램을 종료시키는 기능)

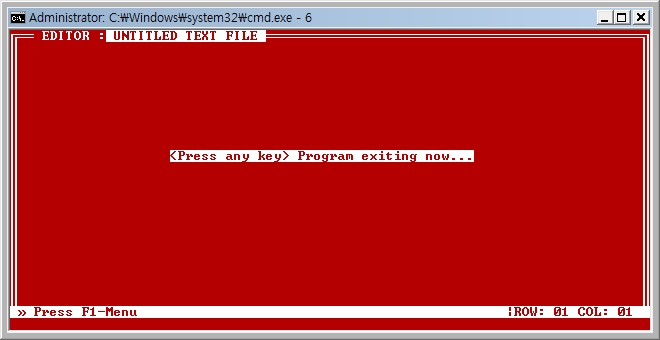


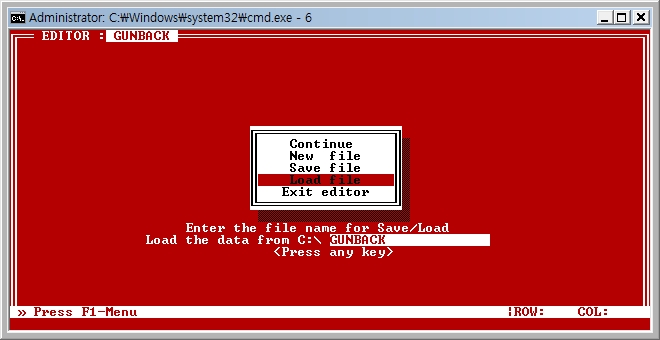
1. 행,열 표시기능 (화면의 오른쪽 하단에 현재 커서의 행,열의 위치를 표시해줌)

커서의 위치에 따라 변하는 ROW COL의 값을 표시해줌



1. 문서의 제목표시 기능   
   (화면 상단에 현재 편집 하고있는 문서의 제목을 표시해줌, 세이브, 로드시에 제목변경)





현재 제목을 표시해주는 표기능

**6. 후기**

한 학기 동안 어셈블리어 강의를 들으면서 배운 내용을 모두 쏟아 부어서 무슨 프로그램을 만드는 이런 일을 처음 해봤는데 그 동안 배운 내용을 복습한다는 의미도 있고 저의 어셈블리 프로그래밍의 실력을 테스트 할 수 있는 경험이었습니다.

무엇보다 최종과제를 통해서 지금까지와는 다르게 실제로 사용할 수 있는 프로그램을 만들었다는 게 너무 뿌듯하였습니다. 또한 C++, C같은 고급언어와는 다르게 아주 기초적인 컴퓨터 조작을 이용해서 이런 프로그램을 만들 수 있다는 것과 저급언어를 통해서 컴퓨터가 프로그램을 어떻게 실행하고 그 프로그램은 어떤 방식으로 컴퓨터를 제어하는지 컴퓨터에 대한 구조를 알 수 있어서 앞으로 저의 학업에 있어서 좋은 시간이었던 것 같습니다.

최종과제를 처음 만들 때 학기 중에 과제로 하였던 2줄짜리 에디터를 토대로 만들었고 도스에디터를 표본으로 삶고 만들었습니다. 그래서 약간은 도스에디터와 닮은 모습입니다. 코딩을 시작하기 전에 에디터에 첨가할 기능들을 정하고 그 기능들이 어떤 순서로 어떻게 실행될지 계획하기 위해 플로우 차트로 프로그램의 흐름을 만들어놓고 시작하였습니다. 또한 예전에는 코딩을 할 때 코딩을 전부 끝낸 뒤에 주석을 달았던 버릇이 있었는데 어셈블리어를 하다 보니 코드가 너무 길어져서 이젠 코딩 하기 전에 주석으로 코드가 들어갈 자리에 어떤 기능을 하는 코드가 들어갈지 먼저 써놓고 코딩을 시작하게 되었습니다.

최종과제는 생각보다 쉽지 않아서 처음에 계획했던 플로우차트 대로 구현이 힘든 것도 있고 생각 했던 기능이 구현하기 힘들어서 삭제하고 또 중간에 새로운 기능에 대한 아이디어가 생각나서 추가한 것도 있었습니다.

초기에 플로우차트를 만들 때는 완벽하게 계획했다고 생각했지만 나중에 코딩을 하면서 보니 굉장히 만은 부분에서 첨삭, 수정하는 부분이 생긴다는 걸 깨달았습니다. 또한 처음에 생각해야 할 것들을 알 수 있었습니다.

그리고 프로시져가 어떤 기능을 수행할 때 예외상황, 예를 들면 화살표가 벽에 부딪히면 이전 / 다음줄로 이동해야 된다는 것 같은 예외상황을 반드시 생각하면서 프로그래밍을 짜야 좋은 프로그래밍이 된다는 것과 항상 버그는 존재하고 코딩하는 시간보다 디버깅을 하는 시간이 오래 걸린다는 것을 아주 잘 알 수 있었던 시간 이었습니다.

저의 프로그램을 보강할 시간이 있다면 문서의 영역을 한 페이지에서 그치지 않고 2페이지, 3페이지로 넘어가고 스크롤 되는 기능을 추가하고 옵션으로 글자의 크기, 색상을 변화 시킬 수 있는 기능과 삽입/수정모드를 변환 할 수 있는 INSERT키의 기능을 추가했으면 하는 아쉬움이 남습니다.

이번 학기에 어셈블리어를 배우면서 내용을 돌아볼 수 있는 시간이어서 좋은 시간이었습니다.

한 학기 동안 고생하신 교수님께 감사말씀을 올립니다.

**7. 참고문헌**

1. IBM PC 어셈블리 프로그래밍, PETER ABEL 저, 2009