Computer Design

~ HW 10 ~

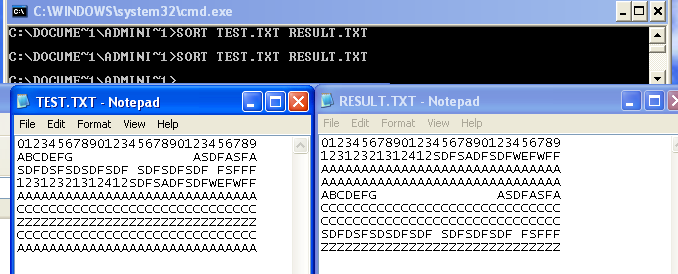
2013210111 남세현

VMWare, Windows XP 32bit MASM 5.1 사용

1. 목표

파일 내 Record들을 Sorting하는 프로그램을 만듭니다.

1. 실행 화면



1. 구현
   1. 입력받은 Parameter를 구분합니다.
      1. PSP에서 source file과 destination file의 이름을 구해냅니다. 두 파일명 사이에 20H가 있으므로 그것으로 구할 수 있습니다.
   2. Int 21 Func 3DH로 Source File를 Open합니다.
   3. Int 21 Func 3CH로 Destination File를 Create합니다.
   4. Int 21 Func 5AH로 임시파일을 생성합니다.
   5. 우선 Source File의 데이터를 Destination File로 옴깁니다.
      1. Int 21 Func 3FH로 Record를 읽고
      2. Int 21 Func 40H로 Destination File’s Handle에 Write합니다.
      3. Int 21 Func 3FH로 읽어들인 Record의 길이가 0이 될때까지 합니다.
   6. Read Handle과 Write Handle이 있습니다. 우선 Read Handle에 Destination File’s Handle, 그리고 Write Handle에 Temporary File’s Handle를 설정합니다.
   7. Buffer는 1번 버퍼, 2번 버퍼가 존재합니다.
   8. 우선 1번 버퍼에 Read Handle로부터 Record를 읽어들입니다.
   9. SI에 1번 버퍼, DI에 2번 버퍼를 설정합니다.
   10. 기초값들을 설정합니다 ( IS\_CHANGE = false, etc,,,)
   11. DI가 가리키는 버퍼에 Read Handle로부터 Record를 읽어들입니다.
   12. SI가 가리키는 버퍼와 DI가 가르키는 버퍼를 비교합니다.

여기서 의도는, 항상 Record를 읽어서 받아들일 버퍼는 DI가 가리키는 버퍼입니다.

오름차순으로 정렬하기 위해서는 DI와 SI가 가리키는 버퍼들을 비교하고 난 후, 항상 SI에 ‘큰’ 버퍼가 존재하도록 해야합니다.

* 1. [ SI + i ] < [ DI + i ]면 SI와 DI를 변경해줍니다.
  2. [ SI + i ] > [ DI + i ]면, 원래 SI와 DI를 Switching해줘야 하는데 Switching해주지 않습니다. 스위칭 하지 않았다는 것은 Bubble Sorting에서 Exchange가 일어났음을 의미합니다. 그러므로 IS\_CHANGE의 값을 TRUE로 바꿔줍니다.
  3. DI가 가리키는 버퍼의 값을 Write Handle에 Write합니다.
  4. Read Handle로부터 읽어들인 Byte가 0이 될 때 까지 J~N을 반복합니다.
     1. 정확하게는, 레코드의 끝은 0D0AH이므로 최대 레코드 사이즈인 32Byte 를 읽어올 때 읽어온 바이트의 양이 0이거나, 읽어온 바이트의 끝부분에 0D0AH가 없으면 종료합니다.
  5. Bubble Sorting에 의한 Exchange가 일어나지 않을 때 까지 수행합니다. 즉, J~N을 모든 레코드 갯수만큼 계속 반복했음에도 IS\_CHANGE가 FALSE이면 종료합니다.
  6. Destination File을 Close해주고 프로그램을 종료합니다.

1. 구현시 힘들었던 점

코드가 굉장히 길어지면서 Condition Jump에서 에러가 발생하였습니다. X86에서는 Condition Jump가 일반 Jump보다 멀리 가지 못한다고 합니다. 그래서

CMP AX, BX

JE A10\_EQUAL

위 같은 코드에서 JE에서 에러가 발생할 경우, 아래처럼 바꿔서 하였습니다.

CMP

JNE A20

JMP A10\_EQUAL

A20:

주의 : HW10.ASM 파일을 어셈블 한 후, Link할 때 실행 파일을 MYSORT.EXE로 생성하게 해주셔야 합니다.