Assignment No. 7

Title :- Sorting

Problem statement :- Write x86 program to sort the list of integers in ascending / descending order. Read the input from text file & write the sorted data back to the same text file using bubble sort.

Objective :- • To understand how to implement bubble sort in ALP.

Outcome :- we will able to study & implement sorting of numbers in ALP.

SW packages & HW apparatus used

editor : gedit
assembler : NASM
Debugger : gdb

Theory :-

Bubble sort :-

It referred to as sinking sort is a simple algorithm that repeatedly steps through the list, compares the adjacent elements & swaps them.

The pass through the list is repeated until the list is sorted. The algorithm which is comparision sort, is named for way smallee or largee element bubble to the top of the list.

example:-

9, 6, 2, 12, 11, 9, 3, 7
6, 9, 2, 12, 11, 9, 3, 7
6, 2, 9, 12, 11, 9, 3, 7
6, 2, 9, 11, 12, 9, 3, 7
6, 2, 9, 11, 9, 12, 3, 7
6, 2, 9, 11, 9, 3, 12, 7

- First pass: 6, 2, 9, 11, 9, 3, 7, (12)

★ similarly other passes will be:

- second pass: 2, 6, 9, 9, 3, 7, (11), (12)
- Third pass: 2, 6, 9, 3, 7, (9), (11), (12)
- Fourth pass: 2, 6, 3, 7, (9), (9), (11), (12)
- Fith pass: (2) (3) (6) (7) (9) (9) (11) (12)

Since we need to stored the sorted array in file we require some system calls:

- Open a file

```
mov rax, 2          ; open syscall
mov rdi, fname1     ; file name
mov rsi, 2          ; file access mode
mov rdx, 0777       ; permission set
syscall
```

- Read a file

```
mov rax, 0          ; Read syscall
mov rdi, [fd_in]    ; file pointer
mov rsi, Buffer     ; Buffer for read
mov rdx, length     ; len of data want to read.
Syscall
```

- write a file:

```
mov rax, 01         ; write syscall
mov rdi, [fd_in]    ; file pointer
mov rsi, Buffer     ; Buffer for write
mov rdx, length     ; len of data want to read
Syscall
```

Close a file :

```
mov rax, 3
mov rdi, [fd_in]
syscall
```

Algorithm :-

i) Staet

ii) SI : offset address
BX : Array size

iii) Outee_loop :
set SI = 1st element of array
set DJ = 2nd element of array

iv) Innee_loop :
compare SI with DI
if SI < DI
Then jmp to SKIP
else
exchange SI with DJ

v) SKIP : set SI to next numbee address
Set DI to next numbee address
if BX != 0
jmp innee_loop
jmp outer_loop

vi) END

Conclusion:

We have learn & implement all the system calls related to file & also sort the given array by reading out the content of the file & write out the sorted list.