Assignment - 7

Title :- Bubble sort array of numbers

Problem defination :-

Write ALP to ~~st~~ sort list of integers in ascending /descending order. Read data from text file & write it back to same text file.

objective :-

i) Understand bubble sort algorithm

ii) Understand deletion, creation & writing to file.

S/W requirements :-

Text editor, Assembler (NASM), linker, debugger (gdb).

Theory :-

A special syscall is required for handling files in assembly as follows

Create file & open if it doesnot exist.

```
mov rax,2            → open syscall.
mov rdi, filename    Read & write mode, create if not
mov rsi, 01020       permission set syscall.
mov rdx, 06660
syscall
mov (fd_in), rax     file pointer goes to
                     rax after opening.
```

Write :

```
mov rax, 01
mov rdi, [fd-in]
mov rsi, buffer
mov rdx, length
syscall
```

close :

```
mov rax, 3
mov rdi, [fd-in]
syscall
```

Delete :

```
mov rax, 87
mov rdi, fname
syscall
```

Algorithm :-

1) mov array size to cl.
2) make rsi point to start of array
3) give value of cl to ch.
4) compare value of value pointed by rsi to value pointed by rsi+1
5) If value at rsi is greater, swap it with rsi+1
~~6) If value at rsi is greater,~~
6) Increament rsi.
7) Decreament ch
8) If ch ≠ 0 goto ④

9) decreament ch
10) If cl≠0, goto ②

1] comparing ([rsi] with [rsi+1])

```
mov ah , [rsi] → mov value at rsi to ah
mov al , [rsi+1] → movvalue at rsi+1 to al
cmp ah,al → compare ah,al.
```

2] Swap values

```
ah has value at rsi
al has value at rsi+1
give location at rsi value of al
give location at rsi+1 value of ah
mov [rsi],al
mov [rsi+1], ah
```

Conclusion :-

Thus we were able to sort a list of numbers obtained from a file & store it back in the file.