----------PROYECTO INDIVIDUAL ARQUITECTURA DE COMPUTADORAS----------

----------NATALIA HERNANDEZ LOPEZ----------

----------PROFESOR. RONNY GARCIA RAMIREZ----------

----------PROYECTO INDIVIDUAL: ENSAMBLADOR DE X86.64----------

Descripción del diseño de software realizado en el proyecto

- ¿Cuáles fueron los principales retos a resolver y cómo se resolvieron?
- ¿Cuáles mejoras se sugieren para el programa y cómo se harían?
- ¿Sus sprincipales conclusiones respecto del los ensambladores?
- Referencias bibliográficas

El proyecto se realizó con la técnica de ordenamiento Bubble sort utilizando las direcciones de cada línea de texto que se encuentran almancenada en un “array de direcciones” llamado long_lineas.

A la hora de empezar el proyecto fue sumamente complicado entender la lógica de funcionamiento de ensamblador x86-64. Sigo siendo una principiante, pero ya entiendo un poco más.

Surgieron muchísimos inconvenientes en el camino.

1. Primeramente el método bubble sort lo estaba implementando con las ubicaciones de los bytes iniciales de cada línea, por ejemplo “la línea 2 inicia en el byte 15” y así con las demás.
    a. Acá surgieron 2 problemas, el primero es que la línea 1 no la estaba guardando, por lo que mi línea 1 me decía que iniciaba donde iniciaba la línea 2. Esto lo solucioné implementanto un test rcx, rcx. Con el fin de siempre almacenar la línea donde está el byte 0.
    b. El segundo problema, el problema que me hizo cambiar la lógica para utilizar mejor el bubble sort, en las imágenes que se presenta en la parte final del documento se ve donde mi manejo con ubicaciones de bytes me llegó a servir dar resultados y para la siguiente ejecución dejó de funcionar. Yo solamente edité el archivo.txt para hacer pruebas de funcionamiento y dejó de funcionar. Simplemente no entendí porque el código .asm nunca lo edité

para este momento. Para solucionar este problema tuve que conversar con un compañero y él me comentó que para ubicar las líneas y cada byte, él estaba utilziando las direcciones de las líneas y que era mucho más sencillo y el tema de la precisión no era tan relevante. Por lo que opté a cambiar de método.

   c. Tristemente no pude realizar el histograma por tema de tiempo, el ordenamiento me consumió el 95% del tiempo y una vez logrado lo tomé como victoria, por lo que no se realizó el histograma.

2. Hubo en particular una parte donde sé que pude haber realizado un proceso más dinámico, pero me di cuenta muy tarde y realizarlo me iba a costar mucho tiempo. Y es que la lógica de buscar un número entre corchetes se utiliza tanto para la lectura y almacenamiento de variables en el buffer Config de config.txt y también se utiliza en el ordenamiento numérico. Técnicamente son las mismas funciones y pude haber utilizado solamente 1 para simplificar el código. Después de esto me gusta el tipo de ordenamiento que utilicé, sé que a lo mejor hice muchos procesos redundantes y que tal vez no eran necesarios, pero para el inicio y funcionamiento considero que logré ambos ordenamientos en muy pocas líneas de código (377)
3. Pensamientos sobre ensamblador. Es muy muy muy complejo e impresionante saber que así se maneja una computadora, el salario que reciben los programadores de bajo nivel tiene todo el sentido del mundo. Es un mundo diferente y complicado. Ya no voy a ver feo a mi computadora cuando va más lento o decide no funcionar. jaja

Referencias

Blum, R. (2005). *Professional Assembly language*. John Wiley & Sons.

Hyde, R. (2021). *The Art of 64-Bit Assembly, Volume 1: x86-64 Machine Organization and Programming*. No Starch Press.

Knuth, D. E. (1998). *Sorting and searching*.

NASM - the Netwide assembler. (s. f.). https://www.nasm.us/doc/

Newest questions. (s. f.). Stack Overflow. https://stackoverflow.com/questions

En las siguientes imágenes muestro el momento donde el programa “muere” de la nada.

Momento de felicidad

```
natih@MSI:~/proyectos/arqui$ vi proyecto.asm
natih@MSI:~/proyectos/arqui$ nasm -f elf64 -g -F dwarf -o proyecto.o proyecto.asm
natih@MSI:~/proyectos/arqui$ ld -o proyecto proyecto.o
natih@MSI:~/proyectos/arqui$ vi proyecto.asm
natih@MSI:~/proyectos/arqui$ nasm -f elf64 -g -F dwarf -o proyecto.o proyecto.asm
natih@MSI:~/proyectos/arqui$ ld -o proyecto proyecto.o
natih@MSI:~/proyectos/arqui$ ./proyecto config.txt archivo.txt
Nota de aprobacion: [43]
Nota de reposicion: [90]
Tamaño de los grupos de notas: [50]
Escala del grafico: [40]
Ordenamiento: alfabetico

Amanda
Beatriz
Camilo
Natalia
Tamara
Xiomara

43
90
50
40
a

Amanda
Beatriz
Camilo
Natalia
Tamara
Xiomara
```

```
natih@MSI:~/proyectos/arqui$ vi archivo.txt
natih@MSI:~/proyectos/arqui$ ./proyecto config.txt archivo.txt
Nota de aprobacion: [43]
Nota de reposicion: [90]
Tamaño de los grupos de notas: [50]
Escala del grafico: [40]
Ordenamiento: alfabetico

Natalia Hernandez [80]
Gisella Lopez [100]
Amanda Patricia [0]

43
90
50
40
a

Amanda Patricia [0]
Gisella Lopez [100]
Natalia Hernandez [80]
```

Y de un pronto a otro

```
natih@MSI:~/proyectos/arqui$ vi archivo.txt
natih@MSI:~/proyectos/arqui$ ./proyecto config.txt archivo.txt
Nota de aprobacion: [43]
Nota de reposicion: [90]
Tamaño de los grupos de notas: [50]
Escala del grafico: [40]
Ordenamiento: alfabetico

Natalia Hernandez [90]
Mariana [30]
AnaBeatriz [49]
Gisella Lopez [50]
Juan Petroleo [20]
Maria [98]

43
90
50
40
a
Segmentation fault (core dumped)
natih@MSI:~/proyectos/arqui$ vi archivo.txt
```

xd