

Tarea 2

- Estudiante: Pablo Alberto Muñoz Hidalgo
- Profesor: Kevin Moraga García
- Curso y Universidad: ITCR Sistemas Operativos
- Año: 2022

Introducción

El problema a resolver en esta ocasión es realizar un tutor de mecanografía en lenguaje ensamblador nasm, en este caso se debe implementar en una llave, esta debe bootear sin necesidad de un sistema operativo, debe presentar el tutor seguido de esto para que el usuario aprenda mecanografía.

Ambiente de desarrollo

Se estará utilizando Ubuntu 20.04.4 LTS y como IDE se utilizará Visual Studio Code. Para probar el booteo se usa Qemu. Además de un repositorio en github.

Estructuras de datos usadas y funciones:

Estructuras:

Macros: Se usan macros para seccionar el programa y utilizar como modulos que pueden recibir y devolver variables, una maravilla.

Registros: Se deben usar registros en NASM.

Strings: Se usan para imprimir la interfaz en pantalla.

Funciones:

Start: Inicia el programa y llama a main.

main: Llama a la función "tutormec".

correcto: En caso de que la tecla sea correcta imprime un mensaje y termina el programa.

Macros:

imprimir: Imprime mensaje en pantalla pedido por el programa.

caracter_de_entrada: Guarda el caracter que ingreso el user, funciona junto a "entrada".

comparador: Compara los caracteres de entrada con los posibles correctos

entrada: Recibe el input del usuario y lo guarda en una variable, llama a comparar e imprime la tecla en pantalla.

salto_linea: Salto de linea básico

limpiar_pantalla: Muchos saltos de linea repetidos para limpiar la pantalla

tutormec: Imprime toda la interfaz y llama a entrada, la cual espera una entrada del usuario. En caso de que entrada sea fallido imprime mensaje de error.

Instrucciones para ejecutar el programa:

1. Inserte la llave USB a una computadora apagada
2. Encienda la computadora
3. Ajuste el orden de booteo para que la llave bootee primero
4. Disfrute del juego

Actividades realizadas por estudiante:

Fecha	Hora de Inicio	Hora de Finalización	Actividad realizada
26/08/2022	12:00 PM	3:00 PM	Investigación de booteo y quemado de usb
26/08/2022	5:00 PM	6:00 PM	Creación del git y descargar add-ons de Visual Studio Code
26/08/2022	6:00 PM	7:00 PM	Booteo efectivo, print de la interfaz básica. Adicionalmente investigación de inputs de usuario.Creación del git y descargar add-ons de Visual Studio Code
26/08/2022	6:00 PM	7:00 PM	Investigar sobre booteo y realización de este, print de la interfaz básica. Adicionalmente investigación de inputs de usuario.
26/08/2022	7:00 PM	7:30 PM	Creación de la documentación.
27/08/2022	10:00 AM	12: 00 PM	Simplificación y organización del código, además de investigación acerca de macros
27/08/2022	1:00 PM	3:00 PM	Modificación de la interfaz gráfica e integración de macros
27/08/2022	7:00 PM	10:00 PM	Integración de input y limpiar pantalla a través de macros
29/08/2022	11:00 AM	12:00 PM	Modificación de la documentación
29/08/2022	12:00 PM	6:00 PM	Implementación de comparar variables y juego completo, a falta de animaciones. Implementar booteo nuevamente
30/08/2022	12:00 PM	1:00 PM	Pruebas de booteo en el LAIMI
30/08/2022	7:00 PM	1:00 AM	Re-hacer el boot, modificar la forma en que se imprime el texto
31/08/2022	11:00 PM	11:30 PM	Modificacion de la documentacion
2/09/2022	8:00 AM	10:00 AM	Modificación de la documentación
2/09/2022	3:00 PM	5:00 PM	Pruebas en el LAIMI

Horas totales: 32 horas

Autoevaluación:

Estado del programa

El programa muestra todo lo necesario para que el usuario aprenda mecanografía, además compara si la letra ingresada es la correcta o no. Sin embargo lo único faltante son las animaciones.

Problemas encontrados y limitaciones adicionales

Tuve un gran problema con las animaciones, no se lograron concretar en el programa, además fue un poco complicado manejar el espacio en memoria ya que solo se podía trabajar con 512 bytes. Por lo demás funciona perfectamente. Tuve problemas con el booteo, sin embargo fueron solucionados a tiempo.

Evaluación

Booteo	Tutormec	Documentación
30/30	40/50	20/20

Reporte de commits:

commit bd67f83a0e27498e77dbc5d1fb7e2996b89142e5

Author: Pablo Munoz pabloamh27@gmail.com

Date: Fri Sep 2 09:06:19 2022 -0600

Documentación e integracion de tar.gz v1.4

commit f44d199d77c696ed7614a09c8ce1a4baa0548ffd

Author: Pablo Munoz pabloamh27@gmail.com

Date: Wed Aug 31 23:10:58 2022 -0600

v1.3 docu actualizada

commit ea43ae6d5df449b243a969e1dc92a6b828ec3374

Author: Pablo Munoz pabloamh27@gmail.com

Date: Tue Aug 30 22:46:18 2022 -0600

Version Funcional v1.2 bootable

commit 7a7f73b4cbe17f5126a78f7588a534a0fea63e3d

Author: Pablo Munoz Hidalgo 53487847+Litecore50@users.noreply.github.com

Date: Mon Aug 29 18:07:55 2022 -0600

Documentacion y comparaciones v1.1

commit 52dfbd14311e3f47a3717da8b905e7f79cb817e9

Author: Pablo Munoz Hidalgo 53487847+Litecore50@users.noreply.github.com

Date: Mon Aug 29 13:30:33 2022 -0600

Versión Funcional 1.0

commit d49438b595eec8f5e603c2725736612956c5de53

Author: Pablo Munoz Hidalgo 53487847+Litecore50@users.noreply.github.com

Date: Fri Aug 26 18:50:45 2022 -0600

Documentación e interfaz básica

commit 61efe9723a2ae99538481cd8bfd72643f7d3bb49

Author: Pablo Munoz Hidalgo 53487847+Litecore50@users.noreply.github.com

Date: Fri Aug 26 17:53:46 2022 -0600

Primeros pasos v0.1

(END)

commit 7a7f73b4cbe17f5126a78f7588a534a0fea63e3d

Author: Pablo Munoz Hidalgo 53487847+Litecore50@users.noreply.github.com

Date: Mon Aug 29 18:07:55 2022 -0600

Documentacion y comparaciones v1.1

commit 52dfbd14311e3f47a3717da8b905e7f79cb817e9

Author: Pablo Munoz Hidalgo 53487847+Litecore50@users.noreply.github.com

Date: Mon Aug 29 13:30:33 2022 -0600

Versión Funcional 1.0

commit d49438b595eec8f5e603c2725736612956c5de53

Author: Pablo Munoz Hidalgo 53487847+Litecore50@users.noreply.github.com

Date: Fri Aug 26 18:50:45 2022 -0600

Documentación e interfaz básica

commit 61efe9723a2ae99538481cd8bfd72643f7d3bb49

Author: Pablo Munoz Hidalgo 53487847+Litecore50@users.noreply.github.com

Date: Fri Aug 26 17:53:46 2022 -0600


Primeros pasos v0.1

Lecciones Aprendidas:

En esta tarea aprendí muchísimo sobre NASM y lo poderoso que es, sin embargo es un lenguaje extremadamente complejo ya que la forma de pensar se me hace muy extraña y difícil. A pesar de todo esto fueron momentos que disfrute ya que se aprende mucho y se sabe lo vulnerable que es todo en la informática. Algo que me fue muy útil fueron los macros ya que estos no ocupan espacio hasta que se usan por lo que es ideal si no se quiere estar modificando el tamaño

del archivo principal y se siente como trabajar con funciones. Por otro lado también me inspiró a aprender como escribir bien :=). PD: Recomendando usar Qemu para probar el booteo.

Bibliografía:

- [1] *Assembly - macros*. (s. f.). Online Tutorials Library. https://www.tutorialspoint.com/assembly_programming/assembly_macros.htm
- [2] *Comparing characters in assembly, nasm*. (s. f.). Stack Overflow. <https://stackoverflow.com/questions/22987212/comparing-characters-in-assembly-nasm>
- [3] *Creating a bootable program in x86 assembly language*. (s. f.). This Could Be Better. <https://thiscouldbebetter.wordpress.com/2011/03/15/creating-a-bootable-program-in-assembly-language/>
- [4] Daedalus Community. (2020, 23 de agosto). *Making an OS (x86) chapter 1 - CPU, assembly, booting* [Video]. YouTube. <https://www.youtube.com/watch?v=MwPjyJ9ulSc>
- [5] *How do I get user input with NASM?* (s. f.). Stack Overflow. <https://stackoverflow.com/questions/523185/how-do-i-get-user-input-with-nasm>
- [6] *How to burn a boot loader onto a USB drive (boot-loader as a .bin)*. (s. f.). Super User. <https://superuser.com/questions/1645622/how-to-burn-a-boot-loader-onto-a-usb-drive-boot-loader-as-a-bin>
- [7] *How to print a number in assembly NASM?* (s. f.). Stack Overflow. <https://stackoverflow.com/questions/8194141/how-to-print-a-number-in-assembly-nasm>
- [8] Rosner, F. (2020, 7 de noviembre). *Writing my own boot loader*. DEV Community . <https://dev.to/frosnerd/writing-my-own-boot-loader-3mld>
- [9] *Which variable size to use (db, dw, dd) with x86 assembly?* (s. f.). Stack Overflow. <https://stackoverflow.com/questions/10168743/which-variable-size-to-use-db-dw-dd-with-x86-assembly>
- [10] *How do I define macros in NASM x86*. (s. f.). Stack Overflow. <https://stackoverflow.com/questions/65935606/how-do-i-define-macros-in-nasm-x86>