

Arquitectura de Computadoras

Trabajo Práctico Especial

Manual de Usuario

Instituto Tecnológico de Buenos Aires



Grupo 22

Integrantes:

- Barnatán, Martín Alejandro (64463)
- Garrós, Celestino (64375)
- Pedemonte Berthoud, Ignacio (64908)
- Weitz, Leo (64365)

Fecha de Entrega: 4 de junio de 2024

2024

Índice:

Índice:	1
Requisitos:	2
Acceso:	2
Comandos:	2
Help.....	2
Registers.....	3
Time.....	3
Eliminator.....	4
Echo.....	5
Clear.....	5
Change Font.....	5
Nano Song.....	5
Test Zero Division.....	6
Invalid Opcode Exception.....	6

Requisitos

En primer lugar, para poder correr nuestro kernel basado en x64 BareBones, el usuario necesita disponer de un entorno de Linux. Por lo tanto, en caso de encontrarse en un dispositivo de sistema operativo Windows, tendrá que utilizar un entorno como WSL. A su vez, se requiere tener el emulador QEMU instalado, así como Docker para la compilación del proyecto.

Acceso

Como primer paso, se debe estar ubicado en la raíz del proyecto. Luego, ejecutando los comandos `./create.sh`, `./compile.sh` y `./run.sh` en ese orden, se podrá acceder al sistema. La primera instrucción crea un contenedor con el nombre `tpe_arqui_1q2024`, en el cual se compilará el proyecto, y descarga la imagen de que este usará. Luego, `compile.sh` inicializa el docker, “limpia” el proyecto (borra todos los archivos `.o`), lo compila y cierra el contenedor. Finalmente, `./run.sh` inicializa el emulador con los argumentos necesarios para su correcto funcionamiento.

Comandos

La shell viene incluida con una serie de comandos que se pueden ejecutar para realizar diversas operaciones. Para cada uno, se debe ingresar su nombre, sin importar mayúsculas dado que no es *case sensitive*. Los comandos son:

- `help`
- `registers`
- `time`
- `eliminator`
- `echo`
- `change_font`
- `nano_song`

La incorrecta introducción de un comando tendrá como resultado un mensaje de comando no encontrado (`command not found`).

Help

El comando `help` permite imprimir en la pantalla un listado con cada comando mencionado anteriormente y su funcionalidad. Una vez ingresado, se obtiene en pantalla:

```

NanoShell $> help
Here's a list of all available commands:
- help --> Help display with all commands
- registers --> Displays the latest backup of registers
- time --> Displays time and date
- eliminator --> Starts the eliminator game
- echo [string] --> Prints the [string] argument in the display
- clear --> clears the display
- change_font --> Changes the current font
- nano_song --> Use command for a surprise
- test_zero_division --> Test for the Zero Division exception
- test_invalid_opcode --> Test for the Invalid Opcode exception
NanoShell $>

```

Registers

Este comando permite obtener un listado del estado de los registros del último backup realizado. La tecla `esc` permite guardar dicho estado a modo de *snapshot*, y se sobrescribe cada vez que se presione. Si al ejecutar el comando no hay ningún backup hecho, se obtendrá un “*Register Status:*” vacío, indicando que se debe realizar el mismo utilizando la tecla correspondiente.

```

NanoShell $> registers
Register Status:
Register backup not done. Press ESC to save register status.
NanoShell $>

```

Si en cambio se realizó el *snapshot* antes de utilizar el comando, se imprimirá un listado de cada registro con su nombre y su valor en dicho momento de la siguiente forma:

```

NanoShell $> registers
Register Status:
RAX: 0
RBX: 1
RCX: 1313503
RDX: 1051256
RSI: 1
RDI: 1
RBP: 1313200
R8: 760
R9: 0
R10: 0
R11: 0
R12: 0
R13: 0
R14: 0
R15: 0
RSP: 1313176
RIP: 1055267
RFLAGS: 0
NanoShell $>

```

Time

Como su nombre indica, con este comando se obtiene en la shell la fecha y hora actual en tiempo de Argentina (UTC-03:00).

```

NanoShell $> time
Current date: 4-6-24
Current time: 0:15:8 hs
NanoShell $>

```

Eliminator

Para poder jugar al juego Eliminator, se debe ejecutar este comando. Lo primero que se obtiene en pantalla es un menú, donde se puede comenzar a jugar de inmediato, configurar el juego o salir. Si se elige la primera opción, el juego comienza por default para dos jugadores.

```
NanoShell $> eliminator
Press space to play, s for settings or q to quit the game.
```

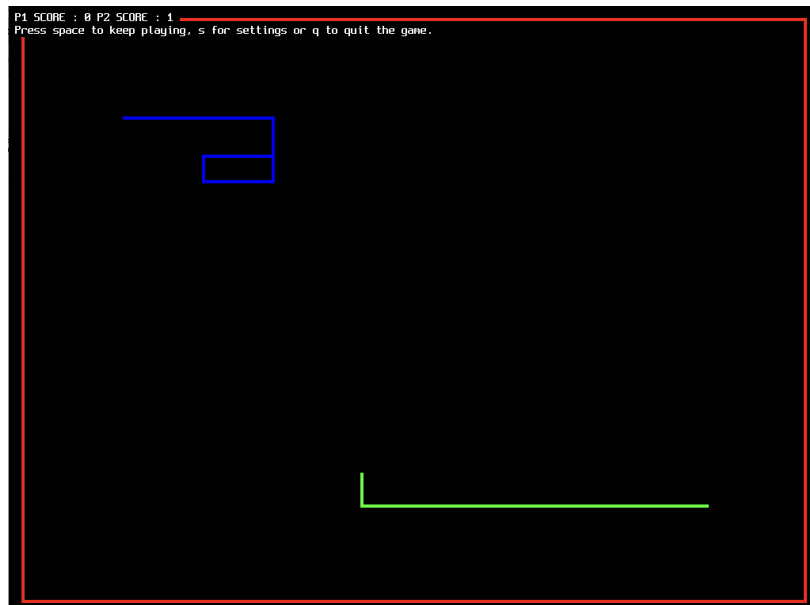
Si se ingresa la tecla *s*, lo primero que se pregunta es si se desea que el juego sea para dos jugadores. Luego, se pide indicar en una escala de 1 a 4 la velocidad que se desee que tengan los usuarios en la partida.

```
NanoShell $> eliminator
Press space to play, s for settings or q to quit the game.
Do you want to play in two player mode? (y/n)
Select your speed from 1 to 4
```

Al elegir la velocidad, automáticamente comienza el juego. En el modo de un sólo jugador, se deben usar las teclas *w*, *s*, *a* y *d* para moverse. El jugador dispondrá de la víbora azul para moverse en la pantalla y jugar. Al perder, el jugador podrá observar en la parte superior izquierda de la pantalla un contador de sus muertes, y se le abrirá el mismo menú que al inicio para elegir si seguir jugando, configurar el juego o salir.



Si se opta por el modo de dos jugadores, el primero será exactamente igual que el modo anterior, y el segundo podrá moverse utilizando las teclas *j*, *k*, *i* y *l*. Nuevamente, al perder alguno de los dos se mostrará el puntaje parcial de cada jugador, junto con el mismo menú de antes.



Si se elige salir del juego mediante la tecla *q*, se volverá nuevamente a la shell vacía y lista para ejecutar el siguiente comando.

Echo

El comando *echo* permite imprimir *strings* en la pantalla de la Shell. Al utilizarlo, todo lo que se ingrese luego (antes de apretar *enter*) será mostrado en la siguiente línea de la Shell. Tiene un límite de 1000 caracteres.

```
NanoShell $> echo Hello World!  
Hello World!  
NanoShell $>
```

Clear

Para poder limpiar el display de la Shell se tiene el comando *clear*, que borra todos los anteriores y comienza de cero en la línea superior. Lo único que hace es actualizar el display, por lo que el backup de registros por ejemplo sigue estando almacenado.

Change Font

Mediante este comando se permite cambiar el tamaño de la letra de la Shell, permitiendo alternar entre uno más grande y otro más pequeño. Al ejecutarlo, se inicia una nueva con el nuevo tamaño de letra.

Nano Song

Al ejecutar este comando, suena por los parlantes de la computadora la canción de “El Nano” interpretada por Melendi en honor al piloto español 2 veces

campeón de la Fórmula 1 Fernando Alonso, y se imprime una foto suya en el display.

Test Zero Division

Para poder hacer un testeo de la excepción de división por cero se encuentra el comando *test_zero_division*. Una vez realizado, se pueden observar los registros como estaban al momento de saltar. Luego, se debe apretar *enter* para volver a la Shell.

```
NanoShell $> test_zero_division
Exception: Cannot Divide By Zero
Register Status:
RAX: 0
RBX: 0
RCX: 17
RDX: 18
RSI: 4207072
RDI: 1312572
RBP: 1313576
R8: 768
R9: 0
R10: 0
R11: 0
R12: 0
R13: 0
R14: 0
R15: 0
RSP: 1313528
RIP: 4194576
RFLAGS: 8
Press ENTER to go back:
```

Invalid Opcode Exception

Con el comando *test_invalid_opcode* se puede realizar la misma operación que en el caso anterior, pero esta vez con la excepción de invalid opcode. Se imprimen los registros y se debe retornar apretando *enter*.

```
NanoShell $> test_invalid_opcode
Exception: Invalid Operation Code
Register Status:
RAX: 0
RBX: 0
RCX: 18
RDX: 19
RSI: 4207091
RDI: 1312612
RBP: 1313616
R8: 768
R9: 0
R10: 0
R11: 0
R12: 0
R13: 0
R14: 0
R15: 0
RSP: 1313568
RIP: 4194580
RFLAGS: 8
Press ENTER to go back:
```