



Instituto Tecnológico de Buenos Aires

*TPE Arquitectura de Computadoras (72.08)*

## **MANUAL DE USUARIO**

Integrantes:

Axel Castro Benza: 62358 - [acastrobenza@itba.edu.ar](mailto:acastrobenza@itba.edu.ar)

Lautaro Farias: 60505 - [lfarias@itba.edu.ar](mailto:lfarias@itba.edu.ar)

Simón Marabi: 61626 - [smarabi@itba.edu.ar](mailto:smarabi@itba.edu.ar)

## Índice:

<b>Seteo de ambiente.....</b>	<b>2</b>
<b>Acceso al sistema.....</b>	<b>3</b>
<b>Ejecución de comandos.....</b>	<b>4</b>
Help.....	5
Dividebyzero.....	5
Invalidopcode.....	6
Inforeg.....	7
Time.....	8
Changefontsize.....	9
Snakes.....	12
Clear.....	17
<b>Aclaración.....</b>	<b>18</b>

## Seteo de ambiente

El usuario deberá contar con QEMU con emulador con el fin de ejecutar este trabajo y con Docker con el fin de compilar el programa. Tras esto, se deberá descargar la imagen a utilizar mediante el siguiente comando desde la terminal de ubuntu:

```
docker pull agodio/itba-so:1.0
```

Tras esto, el usuario tendrá que posicionarse sobre el directorio en el cual se encuentra nuestro trabajo práctico y luego escribir:

```
docker run -d -v ${PWD}:/root -security-opt seccomp:unconfined  
-ti -name NOMBRE agodio/itba-so:1.0
```

Siendo NOMBRE el nombre que tendrá el contenedor.

Luego, se deberán escribir las siguientes líneas de código para poder compilar el trabajo:

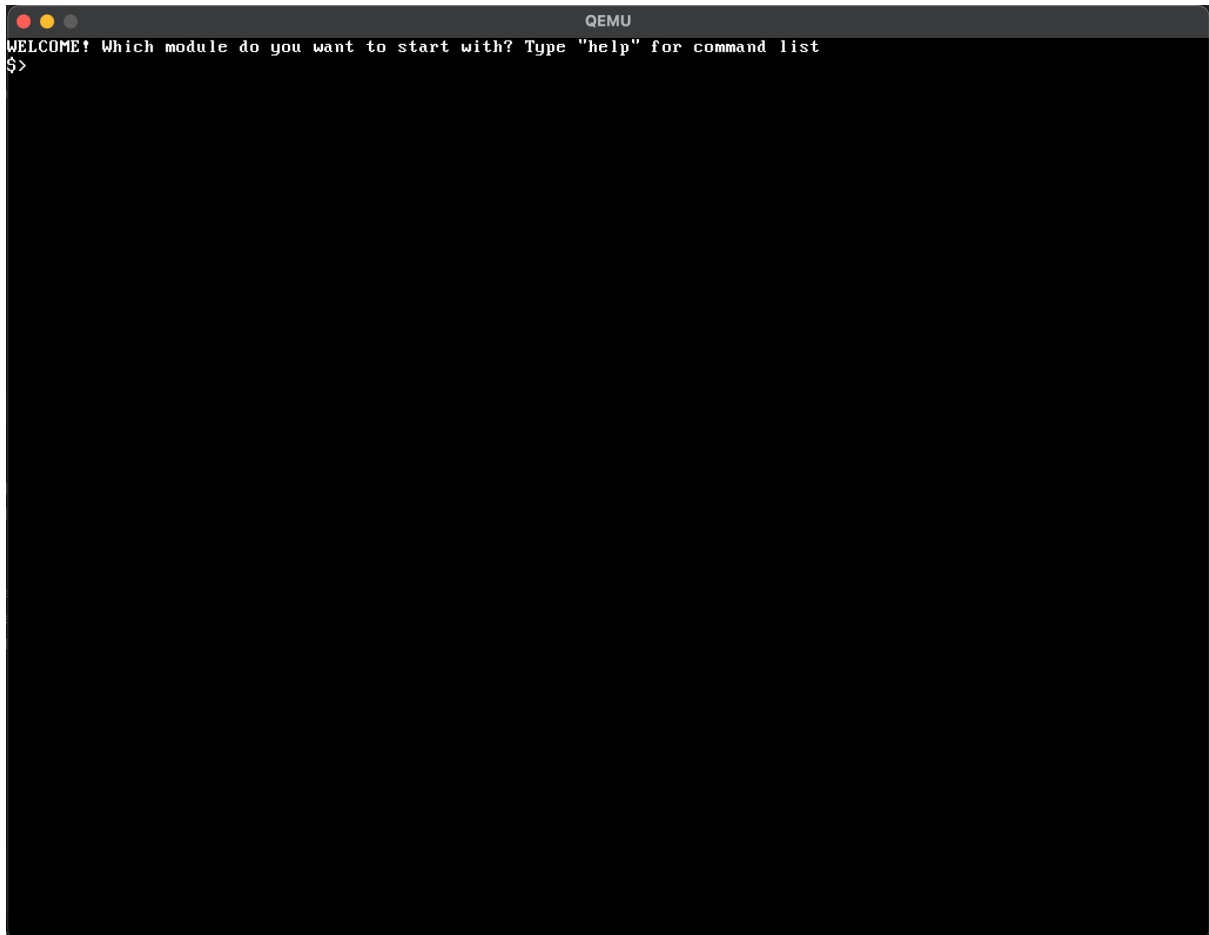
```
cd root  
make all
```

Finalmente, se deberá ejecutar siguiente archivo con el fin de poder visualizar el trabajo práctico:

```
./run.sh
```

## Acceso al sistema

Tras acceder al sistema, se encontrará con la siguiente imagen:



*Imagen 1: Pantalla de bienvenida.*

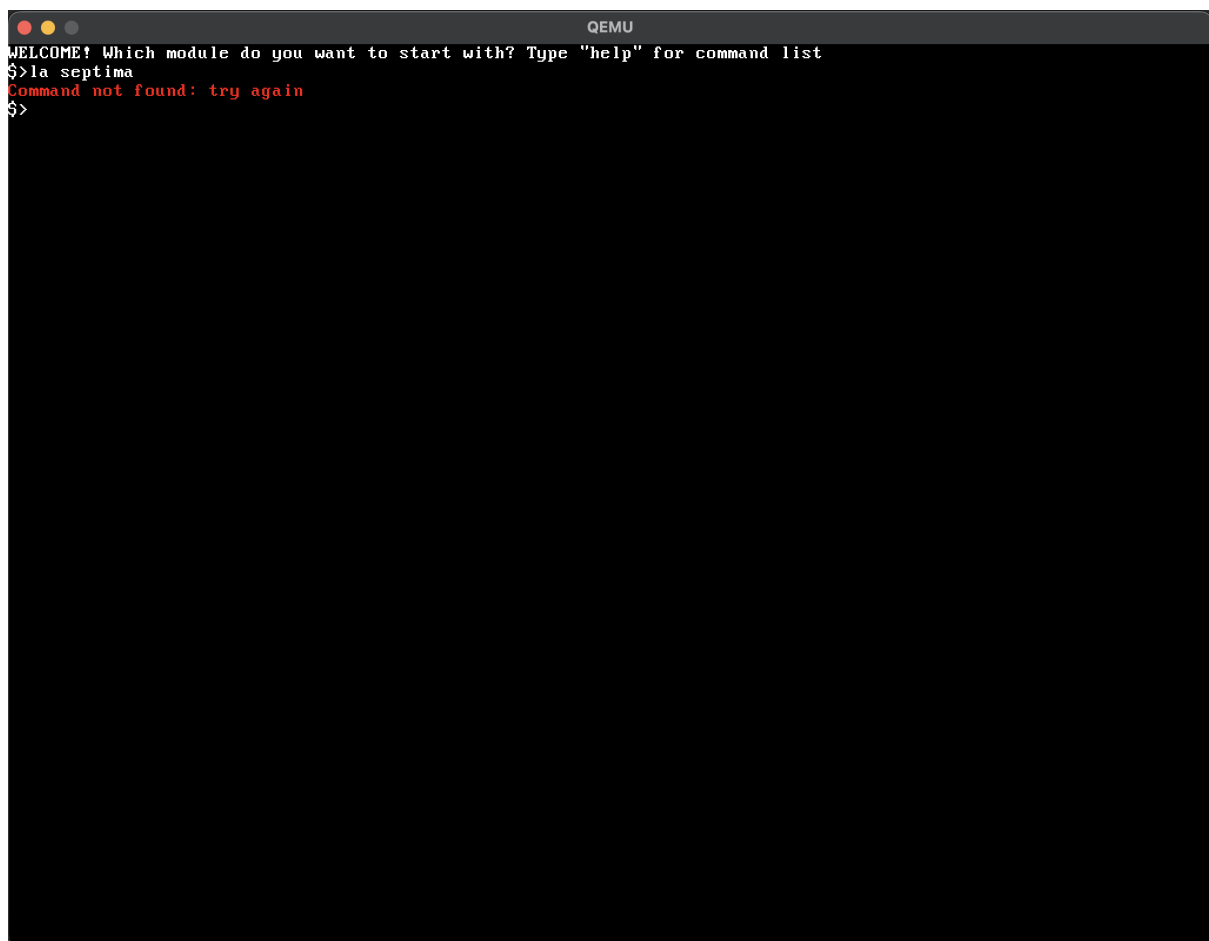
El usuario observará un intérprete de comandos junto con un mensaje de bienvenida. Este intérprete esperará a que se ingrese un comando.

## Ejecución de comandos

Los comandos reconocidos por el intérprete son:

- `help` (Despliega todos los comandos válidos)
- `dividebyzero` (Ejecuta una excepción de tipo división por cero)
- `invalidopcode` (Ejecuta una excepción de tipo código inválido de operación)
- `inforeg` (Imprime los valores de los registros)
- `time` (Imprime la hora del sistema)
- `changefontsize` (Cambia el tamaño de la letra)
- `snakes` (Ejecuta el juego snakes)
- `clear` (Limpia la pantalla)

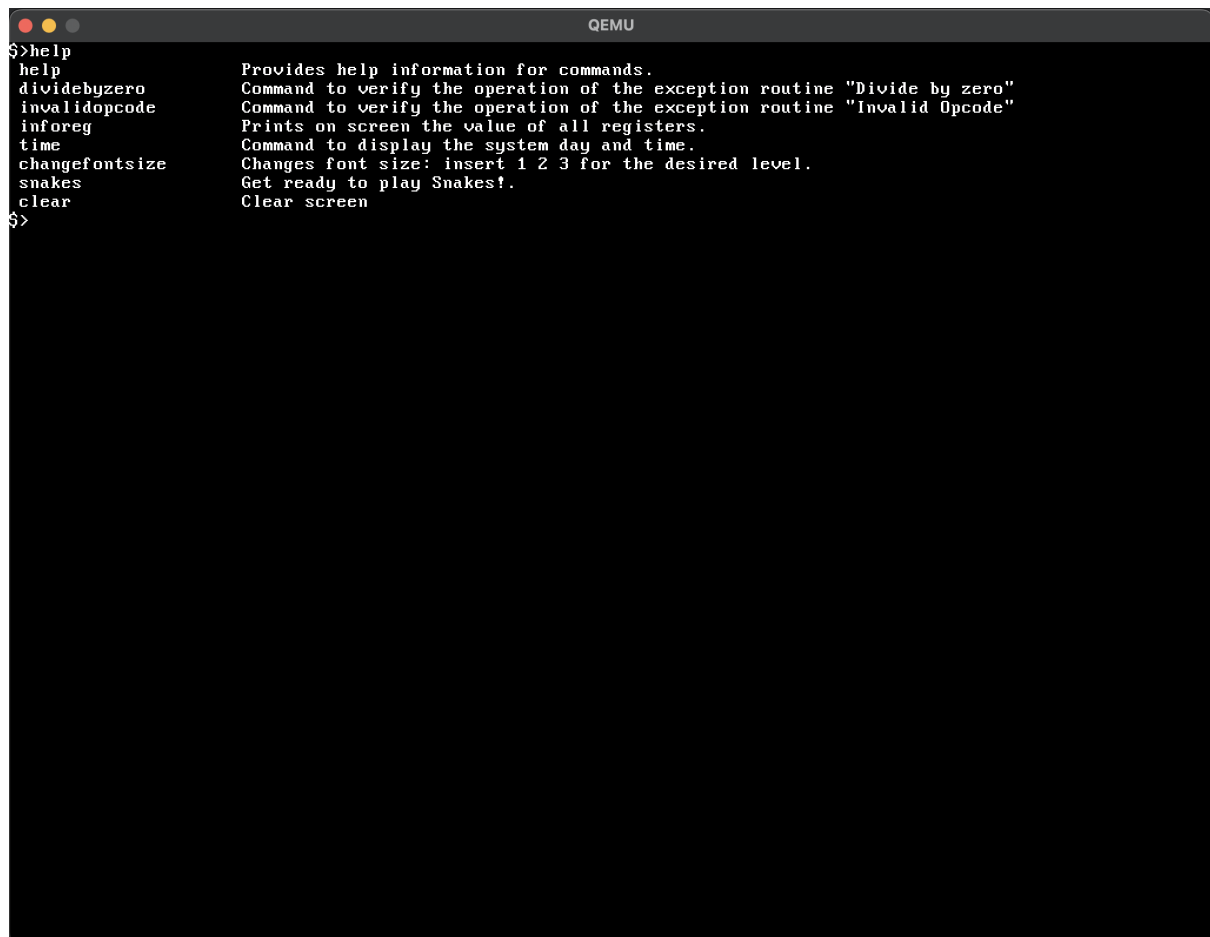
En caso de escribir un comando inválido, se desplegará el siguiente mensaje:



*Imagen 2: Pantalla luego de ejecutar un comando inválido.*

## Help

El comando “help” proporciona información acerca de los comandos válidos junto con una breve descripción de los mismos.

A screenshot of a QEMU terminal window. The window has a title bar with three colored buttons (red, yellow, green) and the text 'QEMU'. The terminal content shows a list of commands and their descriptions. The commands are: help, dividebyzero, invalidopcode, inforeg, time, changefontsize, snakes, and clear. The descriptions are: 'Provides help information for commands.', 'Command to verify the operation of the exception routine "Divide by zero"', 'Command to verify the operation of the exception routine "Invalid Opcode"', 'Prints on screen the value of all registers.', 'Command to display the system day and time.', 'Changes font size: insert 1 2 3 for the desired level.', 'Get ready to play Snakes!', and 'Clear screen'. The prompt '\$>' is visible at the bottom left.

```
$>help
help          Provides help information for commands.
dividebyzero  Command to verify the operation of the exception routine "Divide by zero"
invalidopcode Command to verify the operation of the exception routine "Invalid Opcode"
inforeg       Prints on screen the value of all registers.
time          Command to display the system day and time.
changefontsize Changes font size: insert 1 2 3 for the desired level.
snakes        Get ready to play Snakes!.
clear         Clear screen
$>
```

*Imagen 3: Pantalla luego de ejecutar el comando “help”.*

## Dividebyzero

El comando “dividebyzero” verifica el funcionamiento de la rutina de excepción “Divide By Zero” e imprime los valores de los registros en ese momento. Tras esto, el sistema regresa a la Shell.

```
QEMU
$>dividebyzero
The program aborted due to an exception: Divide by Zero
rip : 0x00000000004000E9
rax : 0x0000000000000000
rbx : 0x0000000000000000
rcx : 0x000000000010DAC0
rdx : 0x0000000000000000
rsi : 0x000000000010DAC0
rdi : 0x0000000000000000
rbp : 0x000000000010B888
rsp : 0x000000000010B808
r8 : 0x0000000000000000
r9 : 0x0000000000000000
r10 : 0x0000000000000000
r11 : 0x0000000000000000
r12 : 0x0000000000000000
r13 : 0x0000000000000000
r14 : 0x0000000000000000
r15 : 0x0000000000000000
rflags : 0x0000000000000000
WELCOME! Which module do you want to start with? Type "help" for command list
$>
```

*Imagen 4: Pantalla luego de ejecutar el comando “dividebyzero”.*

## Invalidopcode

El comando “invalidopcode” verifica el funcionamiento de la rutina de excepción “Invalid Operation Code” e imprime los valores de los registros en ese momento. Tras esto, el sistema regresa a la Shell.

```
QEMU
$>invalidopcode
The program aborted due to an exception: Invalid Operation Code
rip : 0x0000000000000000
rax : 0x0000000000000000
rbx : 0x0000000000000000
rcx : 0x0000000000000000
rdx : 0x0000000000000000
rsi : 0x0000000000000000
rdi : 0x0000000000000000
rbp : 0x0000000000000000
rsp : 0x0000000000000000
r8 : 0x0000000000000000
r9 : 0x0000000000000000
r10 : 0x0000000000000000
r11 : 0x0000000000000000
r12 : 0x0000000000000000
r13 : 0x0000000000000000
r14 : 0x0000000000000000
r15 : 0x0000000000000000
rflags : 0x0000000000000000
WELCOME! Which module do you want to start with? Type "help" for command list
$>
```

*Imagen 5: Pantalla luego de ejecutar el comando “invalidopcode”.*

## Inforeg

El comando “inforeg” imprime los valores de los registros al momento que se sacó una screenshot. Para realizar una screenshot se debe presionar el botón CTRL.

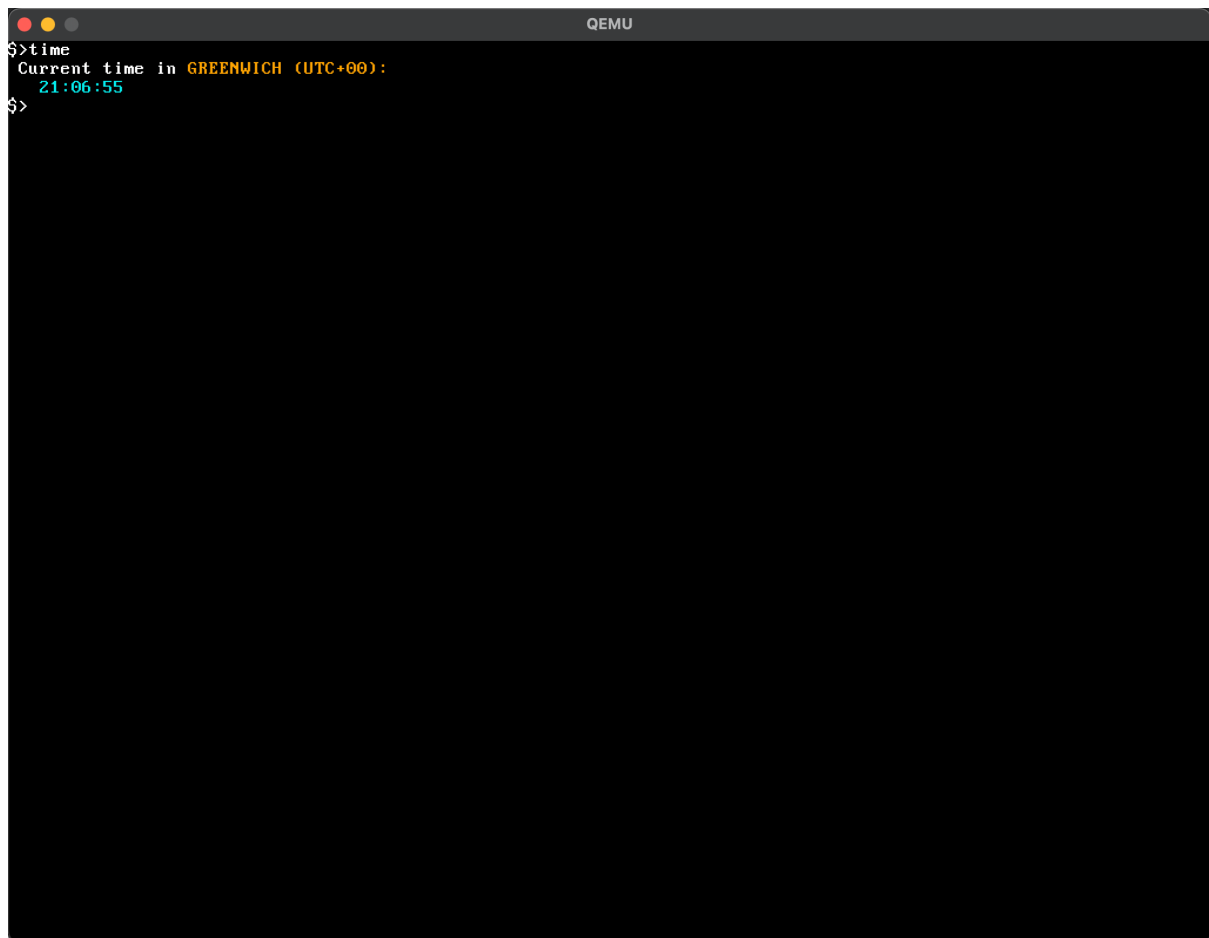


```
QEMU
$>inforeg
A screenshot was not made, press CTRL and try again.
$>inforeg
rip = 0x000000400087
rax = 0x000000000001
rbx = 0x000000000000
rcx = 0x000000000000
rdx = 0x000000000001
rsi = 0x00000010D877
rdi = 0x000000000000
rbp = 0x00000010D878
rsp = 0x00000010D7C0
r8 = 0x000000000000
r9 = 0x000000000000
r10 = 0x000000000000
r11 = 0x000000000000
r12 = 0x000000000000
r13 = 0x000000000000
r14 = 0x000000000000
r15 = 0x000000000000
$>
```

*Imagen 6: Pantalla luego de ejecutar el comando “inforeg”.*

## Time

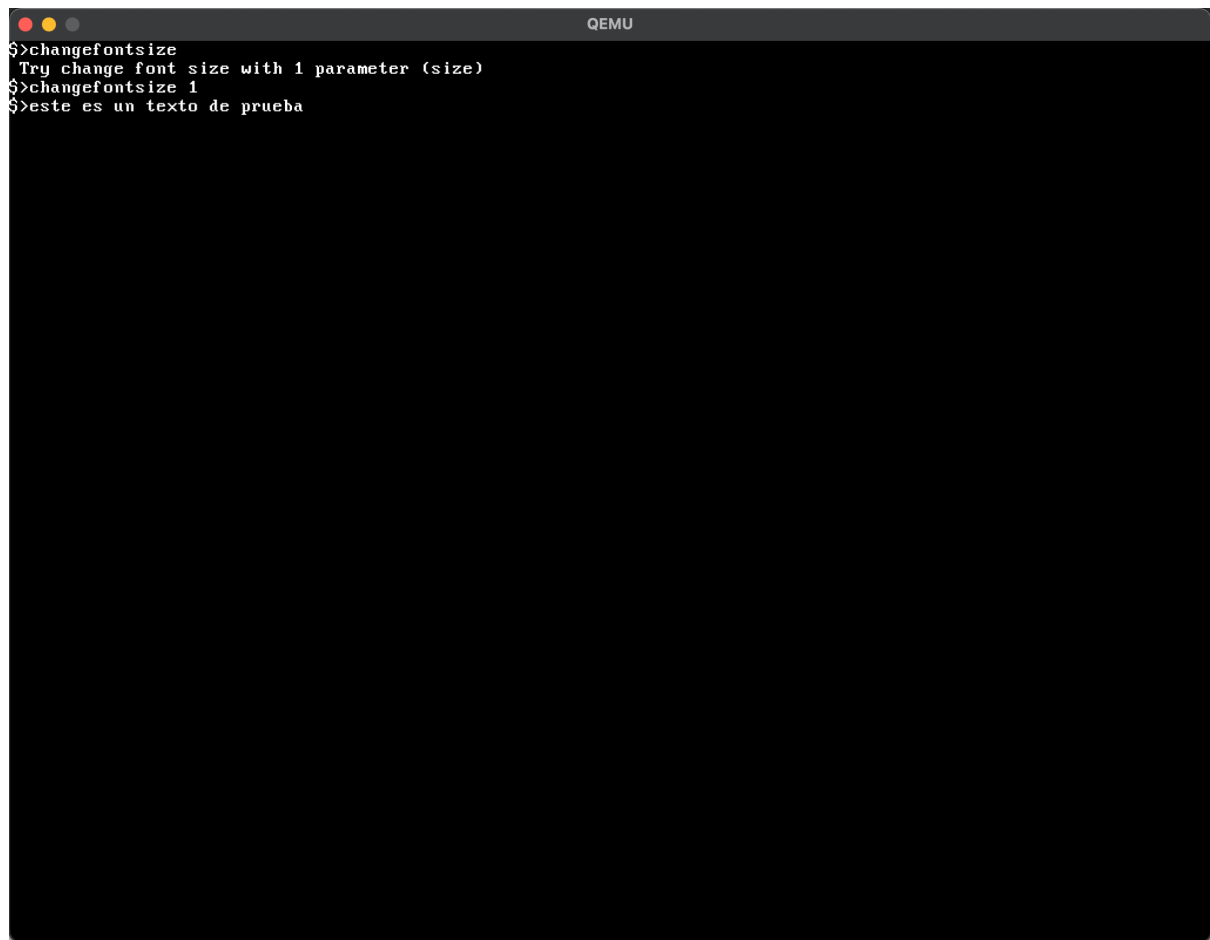
El comando “time” imprime la hora actual del sistema.



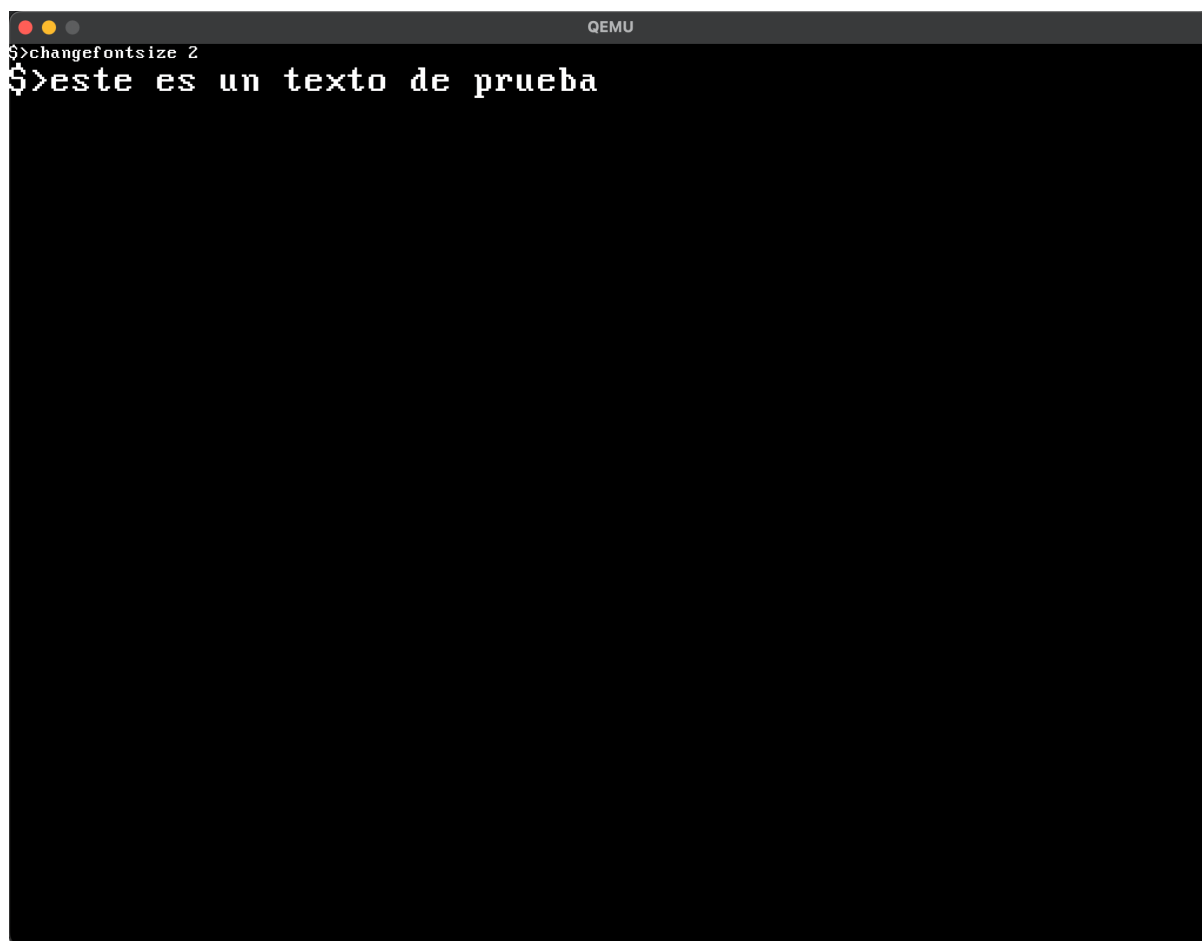
*Imagen 7: Pantalla luego de ejecutar el comando “time”*

## **Changefontsize**

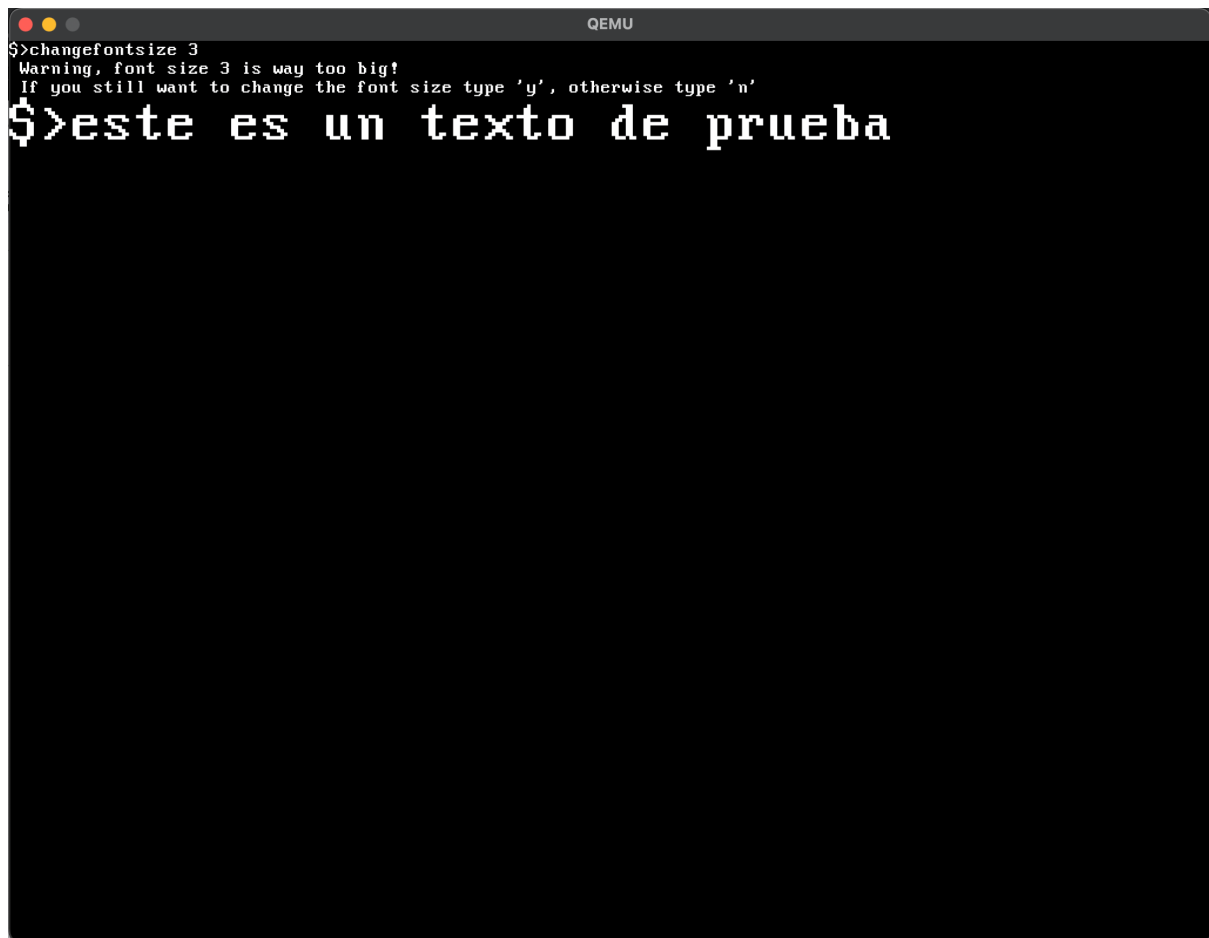
El comando “changefontsize” altera el tamaño de la letra del intérprete de comandos. Hay tres tipos de parámetros válidos: 1 (default), 2 y 3. Siendo 1 el de menor tamaño y 3 el de mayor tamaño.



*Imagen 8: Pantalla luego de ejecutar el comando “changelinesize” con parámetro 1.*



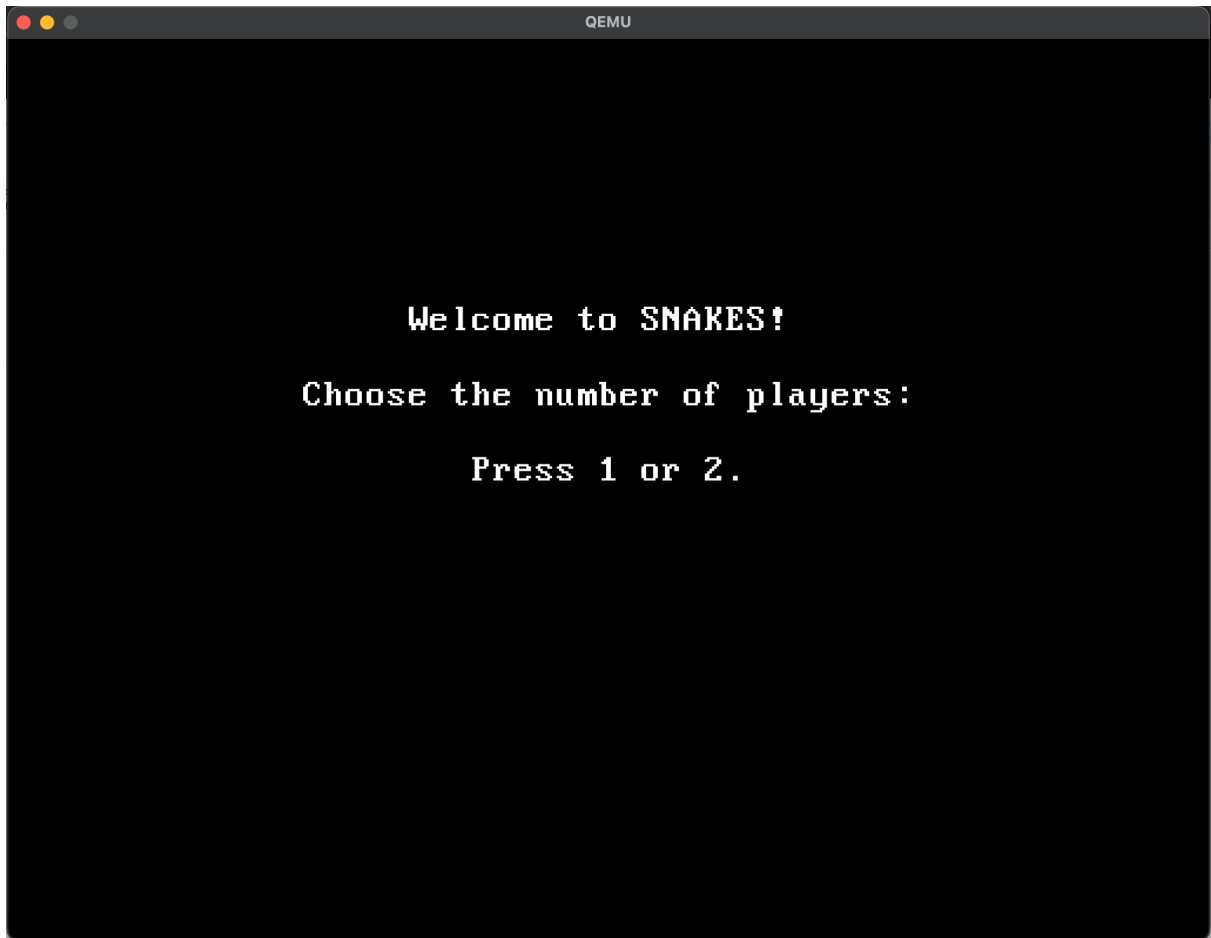
*Imagen 9: Pantalla luego de ejecutar el comando “changelinesize” con parámetro 2.*



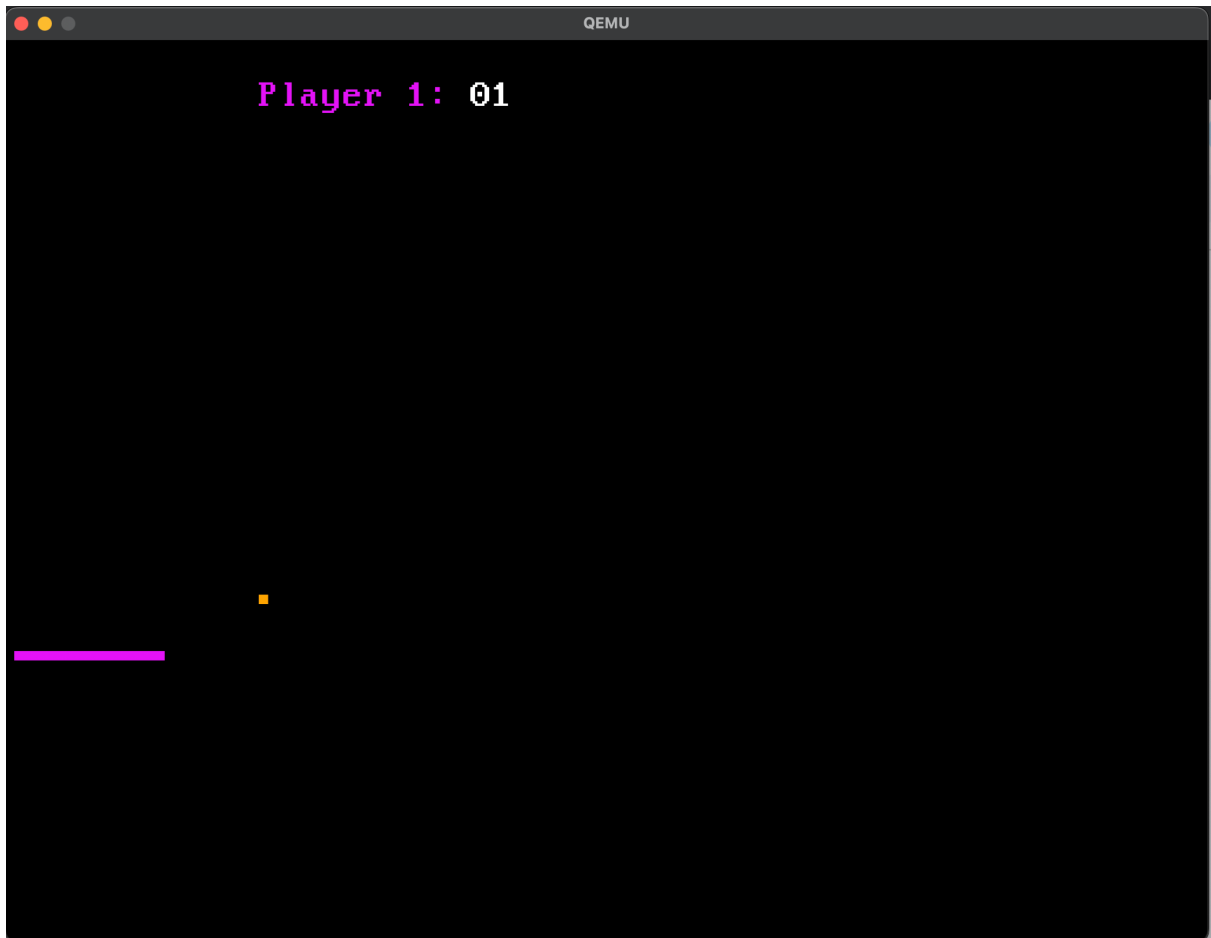
*Imagen 10: Pantalla luego de ejecutar el comando “changepointsize” con parámetro 3.*

## Snakes

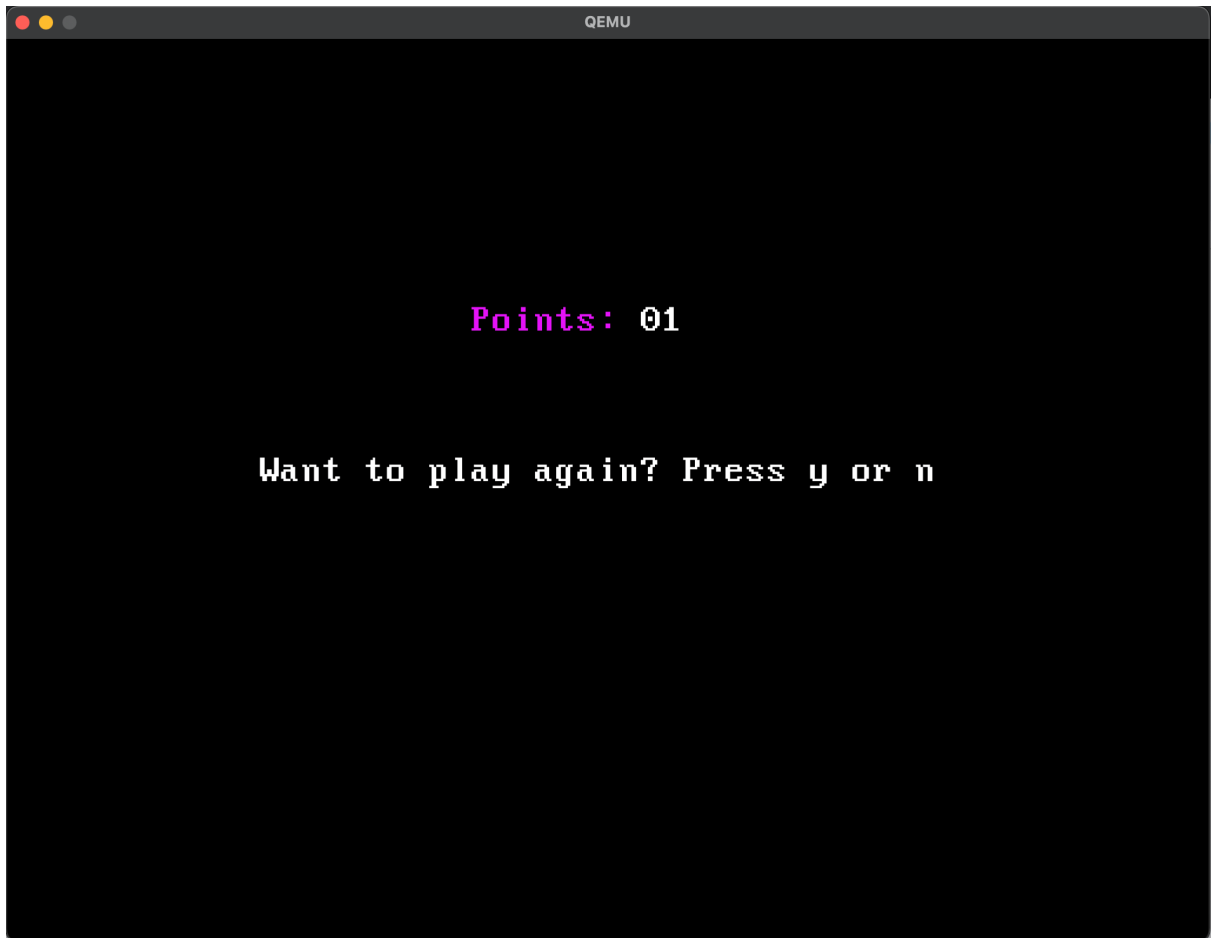
El comando “snakes” ejecuta el juego snakes, donde se puede jugar de a 1 o 2 jugadores. En caso de jugar de a 1, al finalizar el juego se muestra el puntaje del jugador mientras que en caso de jugar de a 2, se muestra el jugador ganador.



*Imagen 11: Pantalla luego de ejecutar el comando “snakes”.*



*Imagen 12: Pantalla luego de iniciar el juego snakes con 1 jugador.*

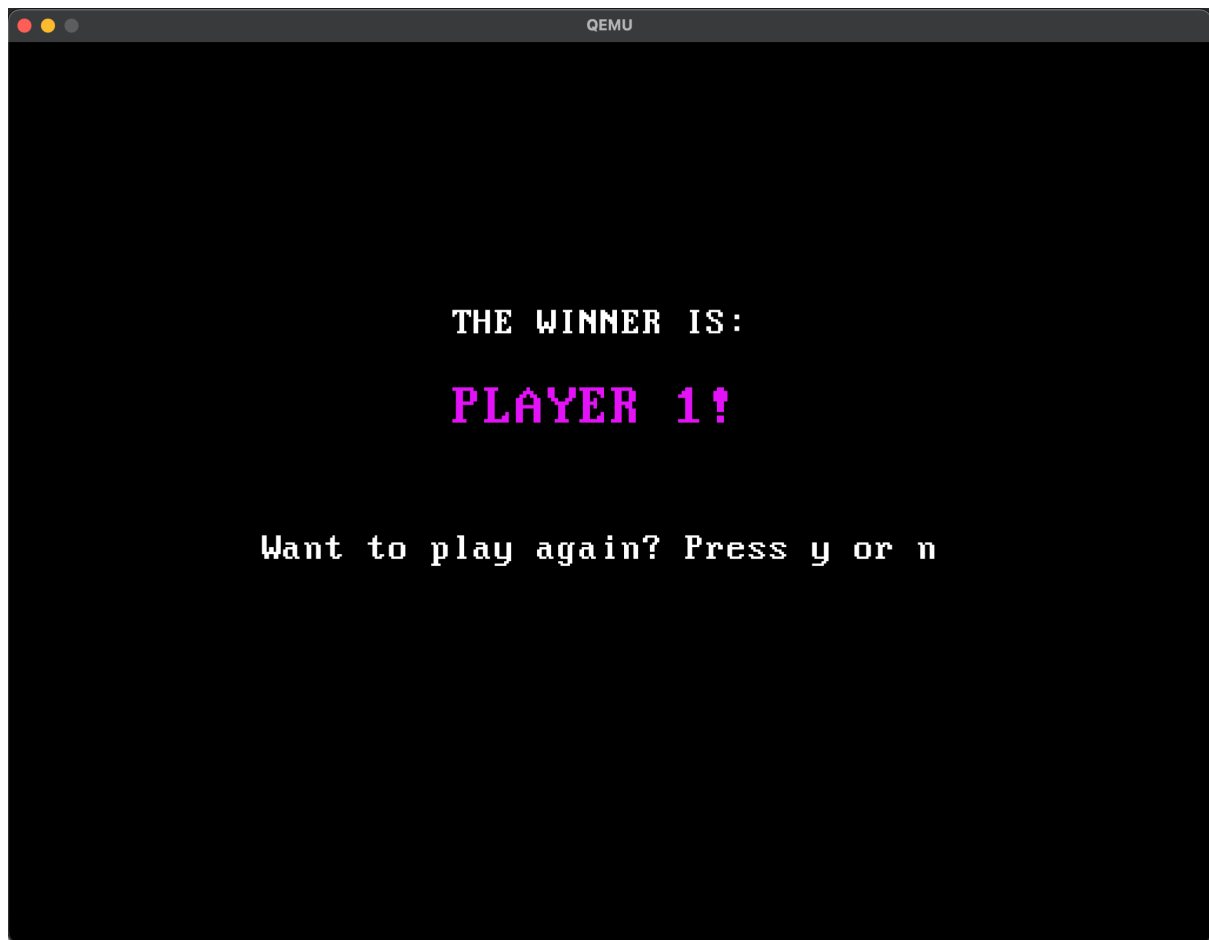


*Imagen 13: Pantalla luego de finalizar el juego snakes con 1 jugador.*





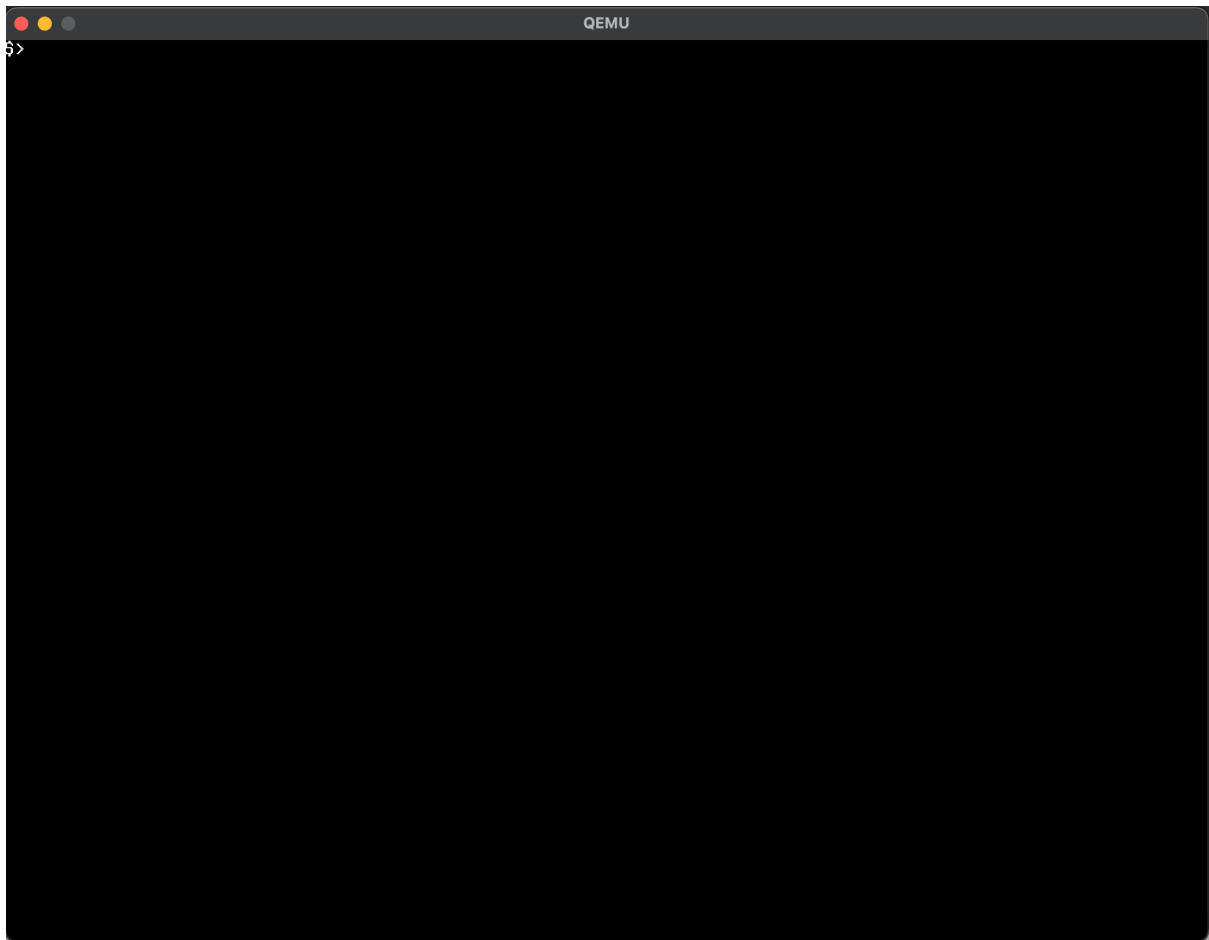
*Imagen 14: Pantalla luego de iniciar el juego snakes con 2 jugadores.*



*Imagen 15: Pantalla luego de finalizar el juego snakes con 2 jugadores.*

## **Clear**

El comando “clear” limpia la pantalla del intérprete de comandos.



*Imagen 16: Pantalla luego de ejecutar el comando “clear”.*

## **Aclaración**

Al ser un grupo conformado por solo alumnos recursantes, se reutilizó mucho código de nuestros trabajos previos. La diferencia más notable fue la adición del juego Snakes y la eliminación de funcionalidades que se requirieron en trabajos previos como por ejemplo la impresión de memoria.