



Instituto Tecnológico
de Buenos Aires

LuccianOS'

Un sistema operativo con buen gusto 🍦

Manual de Usuario - Trabajo Práctico Especial

Arquitectura de las Computadoras (72.08)

Profesores: Santiago Valles, Federico Ramos y Giuliano Scaglioni

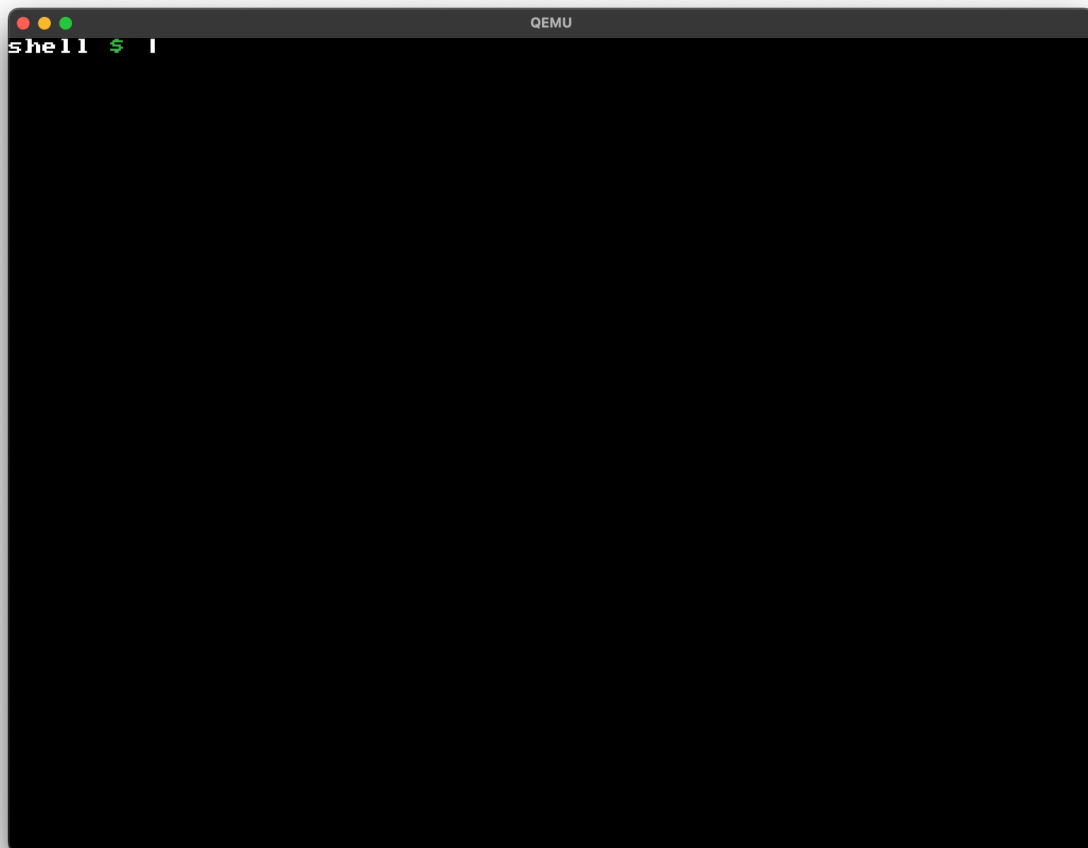
Grupo: 8

Integrantes:

Oliveto, Lucia	loliveto@itba.edu.ar	64646
Pietravallo, Tomas	tpietravallo@itba.edu.ar	64288
Wehncke, Máximo	mwehncke@itba.edu.ar	64018

Sinopsis

Al comenzar, *LuccianOS* abre una terminal de comandos la cual aguarda una entrada por parte del usuario. Si la entrada ingresada no es un comando válido, se mostrará un mensaje detallando lo mismo. Todos los comandos y posibles excepciones regresarán a la consola de comandos.



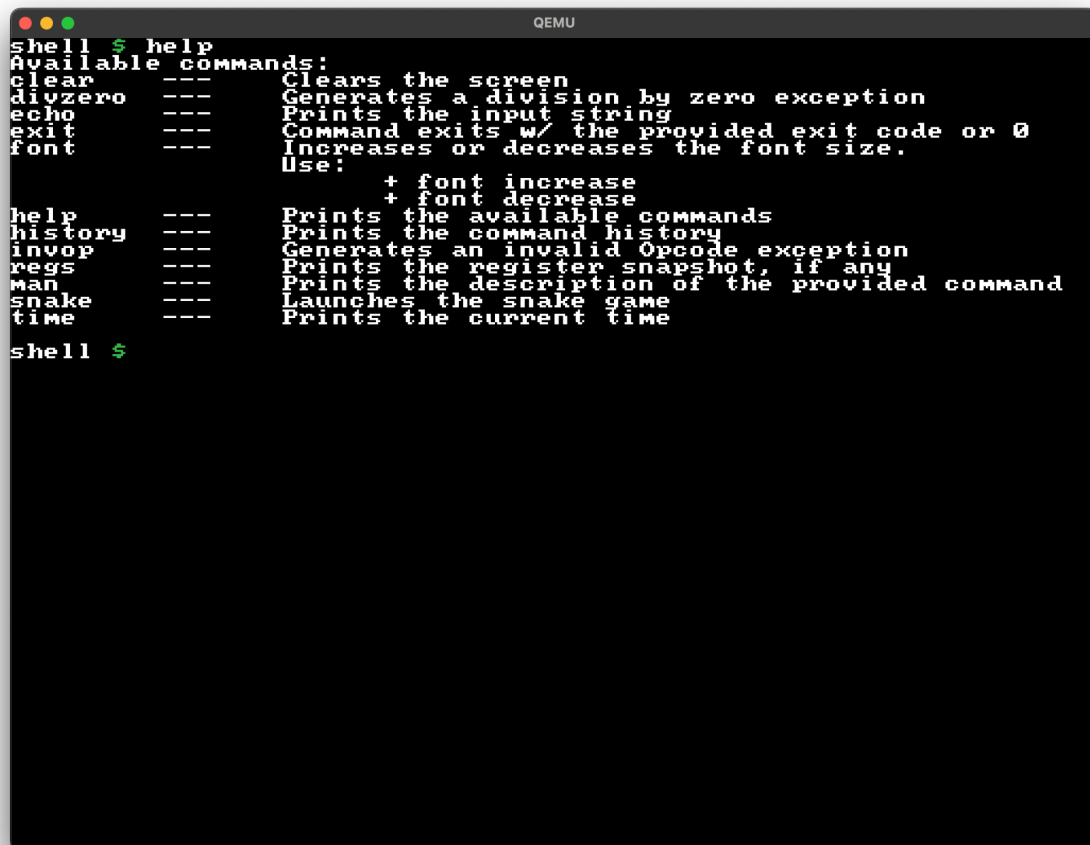
La entrada de la consola está limitada a una longitud máxima de 1024 caracteres – de sobrepasarse se indicará un error. El cursor que se encuentra titilando es el indicador de que puede utilizar el teclado para escribir sobre la consola.

Los comandos podrán, o no, tomar argumentos según se detalle en este manual. Si se proveen argumentos que no corresponden al comando invocado estos serán ignorados. Tanto los argumentos como el comando deben encontrarse separados entre sí por un espacio.

Comandos disponibles

1. *help*

Este comando imprime todos los comandos disponibles, junto a una breve explicación de su propósito.



```
shell $ help
Available commands:
clear      --- Clears the screen
divzero    --- Generates a division by zero exception
echo       --- Prints the input string
exit       --- Command exits w/ the provided exit code or 0
font       --- Increases or decreases the font size.
               Use:
                   + font increase
                   + font decrease
help        --- Prints the available commands
history     --- Prints the command history
invop      --- Generates an invalid Opcode exception
regs       --- Prints the register snapshot, if any
man         --- Prints the description of the provided command
snake      --- Launches the snake game
time        --- Prints the current time
shell $
```

2. *clear*

Limpia la pantalla de la consola de comandos.

3. *divzero*

Genera una excepción de división errónea mediante una división por cero. Esta excepción provocará que la consola de comandos se vea reiniciada.

4. *echo*

Imprime sus argumentos a la consola. Esta entrada puede contener las siguientes secuencias de escape (en negrita):

- **\n** : imprime un salto de línea.
- **\t** : imprime un tabulador horizontal.
- **\r** : retorna el cursor al comienzo de la línea.
- **** : escapa e imprime la barra inversa.
- Códigos de color ANSI de 4 bits de la forma: **\e[X;Ym** donde X e Y son valores de la siguiente tabla:

https://en.wikipedia.org/wiki/ANSI_escape_code#3-bit_and_4-bit.



```
QEMU
shell $ echo \e[31mEste texto es rojo
Este texto es rojo
shell $ echo \e[32mverde, \e[33mamarillo, \e[34mazul
verde, amarillo, azul
shell $ echo \e[30;107mFondo blanco
Fondo blanco
shell $
```

5. *exit*

Este comando retorna con el código de error provisto. El primer argumento provisto deberá ser un número entero no negativo. Si no se provee un argumento se utilizará cero como valor por defecto.

Nota: Esto no posee el efecto de reiniciar la consola de comandos, sólo cambiará el valor de salida del último comando ejecutado por el argumento dado – puede consultarlo mediante el comando “echo \$? ”.

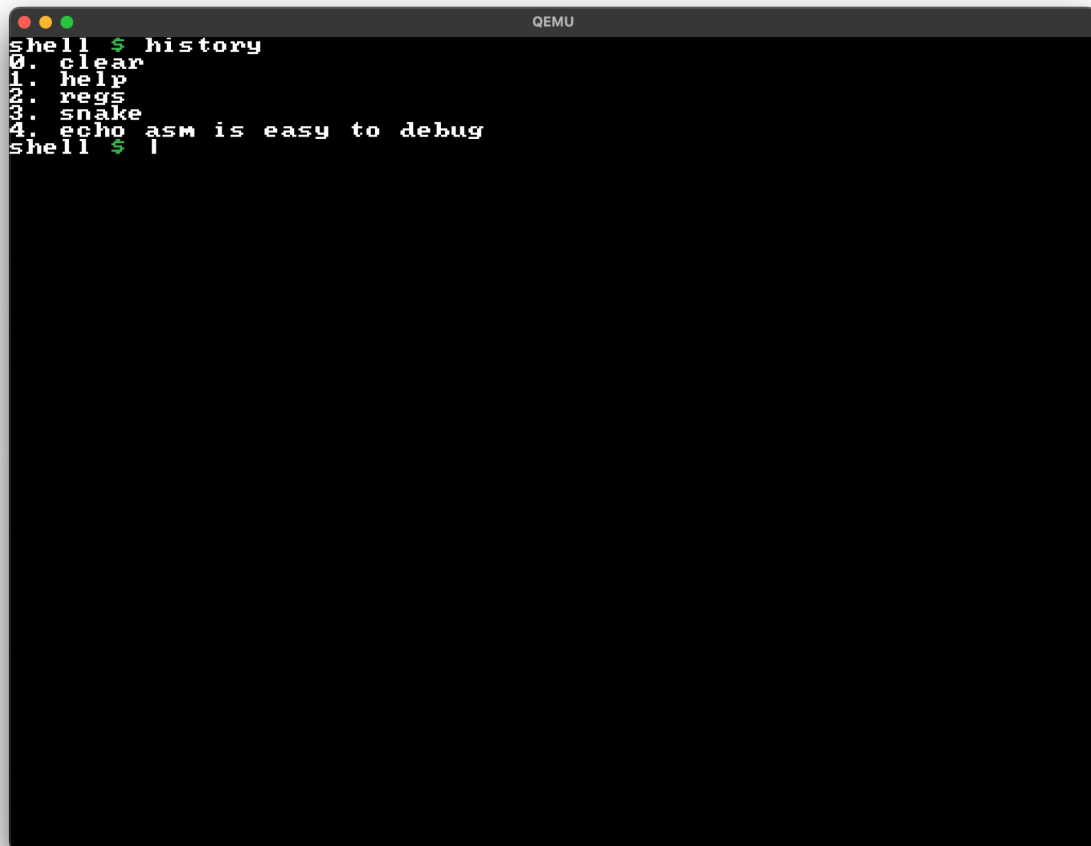
6. *font*

Incrementa o decrementa en una unidad el tamaño de la fuente utilizada por el sistema. Acepta un argumento – “increase” o “decrease” – que determinará el cambio en el tamaño de fuente. Retorna el tamaño de la fuente del sistema luego de los cambios.

Nota: El valor mínimo posible para la fuente es 1, máximo 10, e inicial 2.

7. *history*

Muestra los últimos 10 comandos ingresados a la consola, en orden cronológico inverso. Este historial no se preserva cuando la consola se reinicia a causa de una excepción.



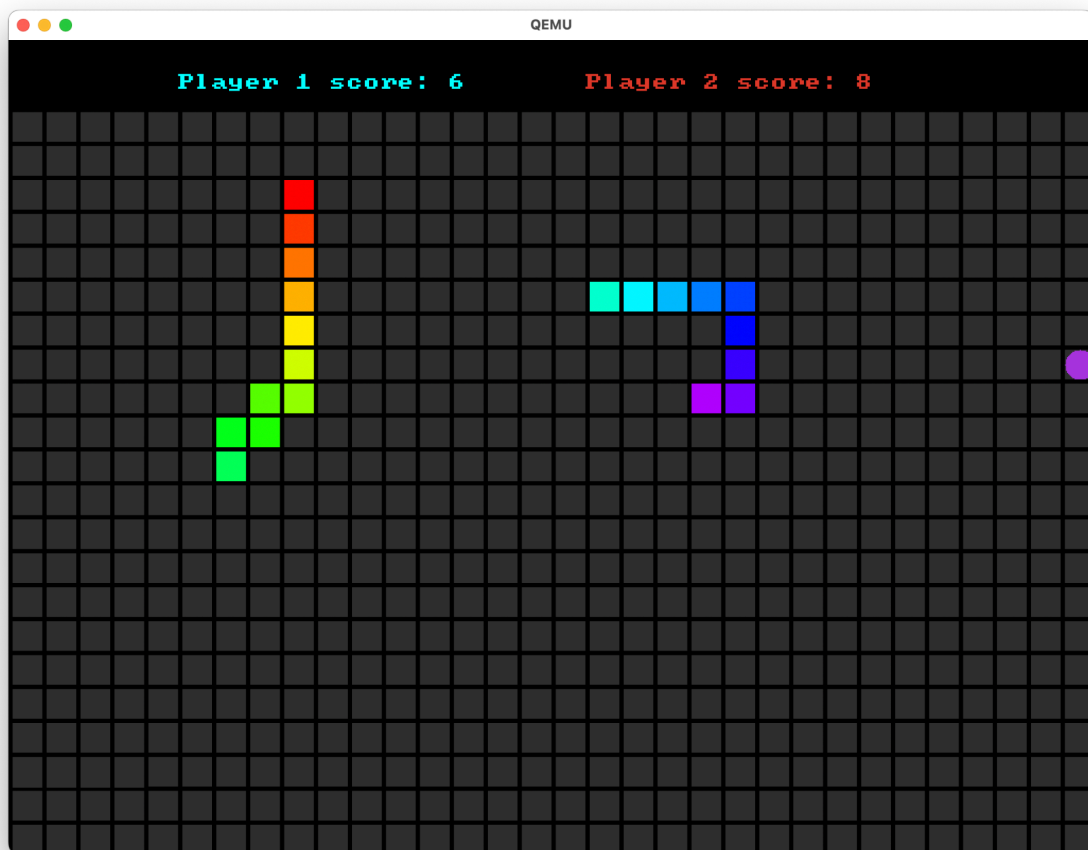
```
shell $ history
0. clear
1. help
2. regs
3. snake
4. echo asm is easy to debug
shell $
```

8. *invop*

Genera una excepción por código de operación inválido. Esta excepción provocará que la consola de comandos se vea reiniciada.

9. *snake*

Lanza el juego de snake. El usuario podrá elegir entre uno y dos jugadores. Las instrucciones del juego se presentarán al correr el comando.



10. *regs*

Imprime el estado actual de los registros. Se ha mapeado la tecla *F12* para en cualquier momento, capturar el estado de los registros al presionarla. Si no se capturó el estado de los registros previo a intentar visualizar su estado, se imprime un mensaje indicando lo mismo.

```
QEMU
shell $ regs
No register snapshot available
shell $ regs
Latest register snapshot:
rax: 5
rbx: 0
rcx: 10DFA7
rdx: 5
rbp: 10DE80
rdi: 3
rsi: 10DFA7
r8: 0
r9: 0
r10: 0
r11: 0
r12: 0
r13: 0
r14: 0
r15: 0
rip: 102452
rflags: 46
shell $ !
```

11. *man*

Imprime una breve descripción del comando pasado como argumento.

12. *time*

Imprime la hora actual del sistema, en formato 24 horas (HHh MMm SSs).