



Instituto Tecnológico de Buenos Aires

TPE - Manual de Usuario

72.08 - Arquitectura de Computadoras

Grupo: HomerOS

Integrantes:

Gonzalo Candisano - Legajo N° 62616

Ben Deyheralde - Legajo N° 63559

Emilio Pablo Neme - Legajo N° 62601

Theo Shlamovitz - Legajo N° 62087

Docentes:

Vallés Santiago Raúl

Merovich Horacio Víctor

Ramos Federico Gabriel

CÓMO EJECUTAR “HomerOS”:

Para compilar y ejecutar el proyecto, es necesario tener docker instalado y la imagen provista por la cátedra (agodio/itba-so). Luego desde la terminal hay que ir al directorio donde se encuentre la carpeta del proyecto (*TPEArqui-CDNS*) y correr el comando: **“./docker.sh”**

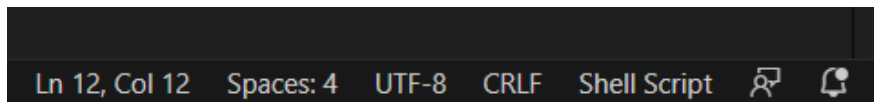
En caso de que suceda este error:

```
run.sh: line 2: $'\r': command not found
run.sh: line 15: syntax error: unexpected end of file
```

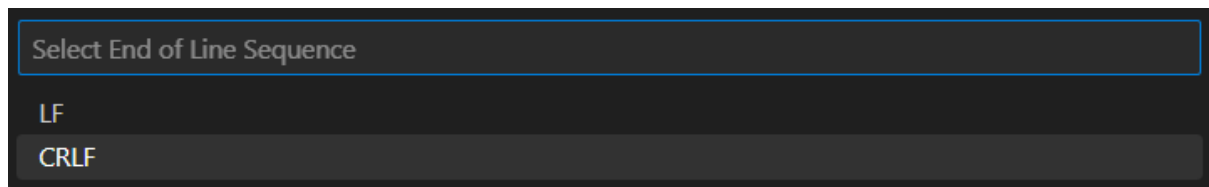
o este:

```
./docker.sh: line 14: syntax error: unexpected end of file
```

Es necesario cambiar de CRLF a LF en el editor de texto que se esté utilizando en el archivo *docker.sh* y el archivo *run.sh*. En este caso fue utilizado el Visual Studio Code y la siguiente imagen se encuentra abajo a la derecha en la pantalla.



Al presionar CRLF, en la parte superior de la pantalla, en el centro, aparece lo siguiente donde hay que presionar LF, guardar ambos archivos y volver a ejecutar *./docker.sh*



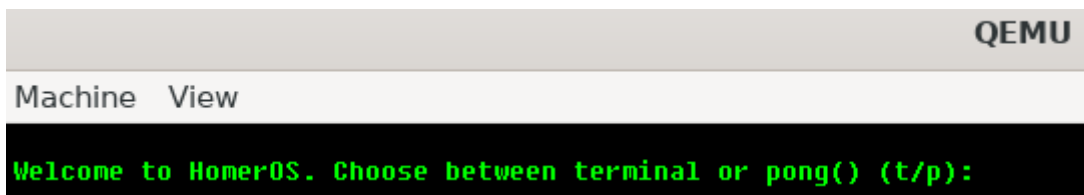
Una vez compilado, se puede volver a ejecutar en cualquier momento utilizando el script *run.sh*, y ejecutandolo con **“./run.sh”**. Del mismo modo, se puede ejecutar **“./run.sh gdb”** para iniciar en modo de DEBUG. En este ultimo caso es necesario abrir otra terminal, moverse al directorio donde se encuentra el trabajo, y ejecutar el comando **“gdb”**. Para la correcta visualización del mismo, es necesario tener el archivo *.gdbinit* en el *%HOME%*.

FUNCIONALIDADES:

Al abrir el qemu, suena la canción de los Simpsons y aparece lo siguiente:

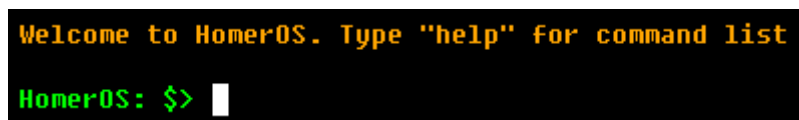


Luego aparece el siguiente texto donde uno elige si quiere ir directo al pong (presionando la tecla “p”) o a la terminal la cual ofrece una variedad de comandos explicados a continuación (presionando la tecla “t”).



Terminal:

En caso de que sea seleccionada la terminal, aparece lo siguiente:



Antes de comenzar a utilizar la terminal cabe destacar que para un uso más fluido de la misma, 2 funciones fueron implementadas.

- Al presionar la tecla “**tab**”, autocompleta el comando si es que lo que está escrito en la terminal es el comienzo de alguno de los que se encuentran en la lista de posibles comandos que aparecen al realizar el comando “**help**” mencionado próximamente.
- Una vez realizado un comando, en el caso de querer repetirlo no es necesario escribirlo de nuevo, tocando la tecla de la **flecha hacia arriba** trae el último comando escrito (solo va un comando hacia arriba). En el caso de arrepentirse de haber utilizado la flecha hacia arriba, se puede presionar la **flecha hacia abajo**, la cual borra lo que está escrito.

En el caso de escribir un comando el cual no se encuentra en la lista de comandos la cual muestra el comando “*help*” sale un mensaje de error indicando al usuario que escriba el comando “*help*” para ver los comandos disponibles que hay.

```
HomerOS: $> a
Command not found. Type "help" for command list
```

Como indica la terminal, al escribir el comando “*help*” se muestra en pantalla una lista de comandos posibles a ejecutar con una breve descripción de cada uno.

```
HomerOS: $> help
Comandos disponibles:

help: gives you a list of all existants commands.
time: prints the time of the OS.
date: prints the date of the OS.
registers: print the state of the registers at the time you screenshot them with CTRL key.
fillregs: fill the registers with stepped values for testing.
div0: divide by zero to trigger exception
invalidop: trigger invalid operation code exception
pong: go to play the "pong" game.
clear: clears the OS screen.
```

El comando “*time*” muestra en pantalla el horario actual del sistema operativo (en formato HH:MM:SS).

```
HomerOS: $> time
Time of OS: 13:54:09
```

El comando “*date*” muestra en pantalla la fecha actual del sistema operativo (en el formato DD/MM/YY).

```
HomerOS: $> date
Date of OS: 05/06/23
```

El comando “*registers*” imprime una lista de los registros con su contenido el cual se carga al hacer un “screenshot”. Este se realiza presionando la tecla “*CTRL*”.

```
HomerOS: $> registers

RAX  0x00000000FFFFFFFF
RBX  0x00000000FF000000
RCX  0x00000000EEEEEEEE
RDX  0x00000000DDDDDDDD
RSI  0x00000000CCCCCCCC
RDI  0x00000000BBBBBBBB
RBP  0x00000000010D700
RSP  0x00000000010D700
R8    0x00000000AAAAAAAA
R9    0x0000000099999999
R10   0x0000000088888888
R11   0x0000000077777777
R12   0x0000000077770000
R13   0x0000000077000000
R14   0x0000000055566666
R15   0x0000000018CF7EB9
RIP   0x000000000400156
```

La función del comando “*fillregs*” es setear un valor alto en los registros e ir decrementandolos en orden para poder realizar un “screenshot” de los mismos durante el proceso y luego imprimirlos con el comando “*registers*” donde aparecen en orden descendiente.

```
HomerOS: $> fillregs

Filling registers...
Press CTRL to save them.
```

El comando “*div0*” dispara la excepción de división por cero, muestra los registros en el momento de la interrupción y reinicia la terminal. Esto lo logramos restaurando los registros originales, guardados luego de la introducción del SO en el Kernel.

```
HomerOS: $> div0

Cannot divide by zero.

RIP 00000000004000E1
RAX 0000000000000003
RBX 0000000000000000
RCX 0000000000000004
RDX 0000000000000005
RSI 0000000000000006
RDI 0000000000000007
RBP 000000000010D780
RSP 000000000010D780
R8 0000000000000009
R9 000000000000000A
R10 000000000000000B
R11 000000000000000C
R12 000000000000000D
R13 000000000000000E
R14 000000000000000F
R15 0000000000000010
RFLAGS 0000000000000202

Welcome to HomerOS. Choose between terminal or pong() (t/p):
```

El comando “*invalidop*” dispara la excepción de realizar una operación inválida, muestra los registros en el momento de la interrupción y reinicia la terminal, de la misma forma que el *div0*.

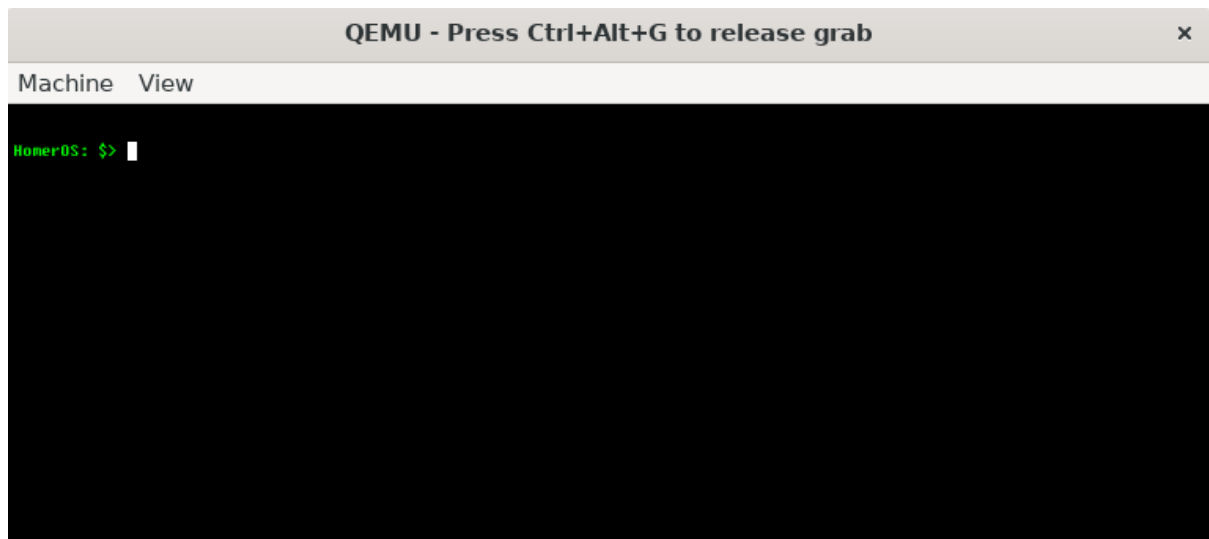
```
HomerOS: $> invalidop

Invalid opcode.

RIP 00000000004000E5
RAX 0000000000000000
RBX 0000000000000000
RCX 0000000000000009
RDX 0000000000000000
RSI 000000000042A327
RDI 000000000042A327
RBP 000000000010D780
RSP 000000000010D788
R8 0000000000000000
R9 0000000000000000
R10 000000000000000B
R11 000000000000000C
R12 0000000000000000
R13 0000000000000000
R14 000000000000000F
R15 0000000000000000
RFLAGS 0000000000000202

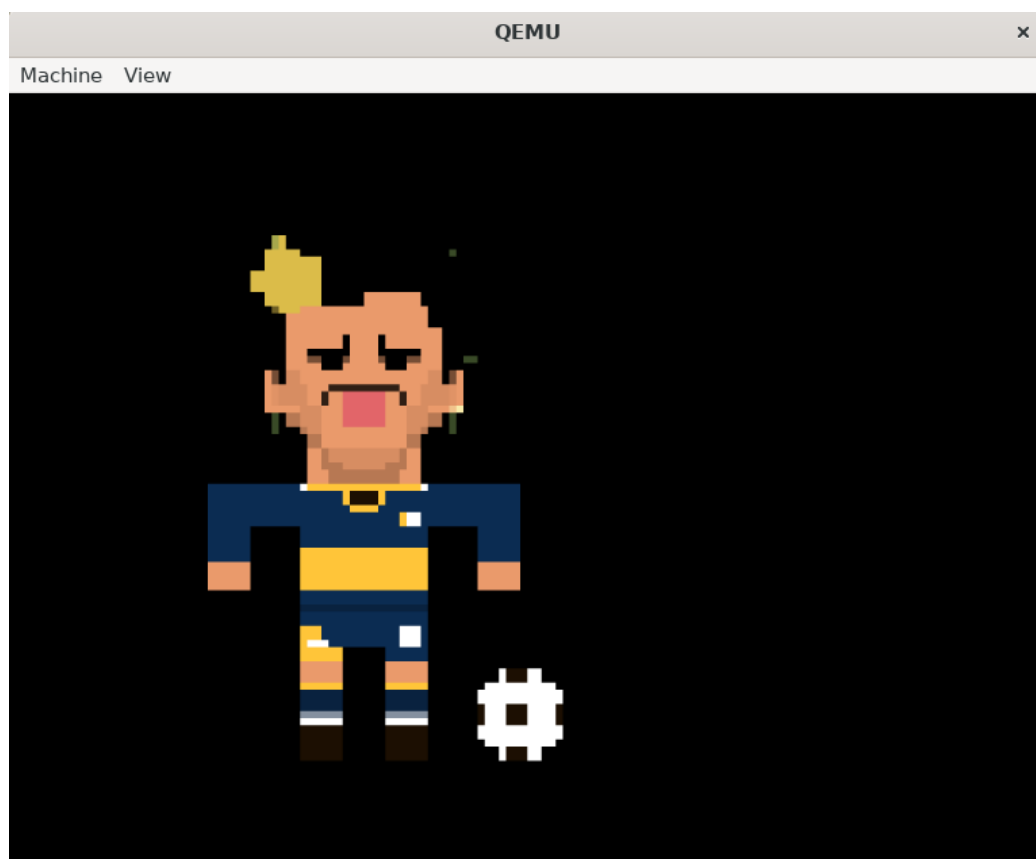
Welcome to HomerOS. Choose between terminal or pong() (t/p):
```

El comando **“clear”** se realizó con el objetivo de, en el momento de utilizar la terminal, borrar comandos anteriores realizados, limpiando la pantalla.



El comando **“pong”** lleva al usuario a jugar al juego tal como si al comenzar a utilizar el sistema, hubiera presionado la tecla **“p”** en vez de **“t”**. El juego pong se encuentra explicado a continuación.

Por último, hay un *easter-egg* en la terminal, lo que vendría a ser un comando “secreto”, el cual no se encuentra en la lista de comandos provista por el comando **“help”** que lo que hace es tocar una canción de Boca y mostrar un dibujo de Diego Maradona con la remera de Boca y una pelota de fútbol. Para ejecutar el comando easter-egg escribir en la terminal **“boca”**.

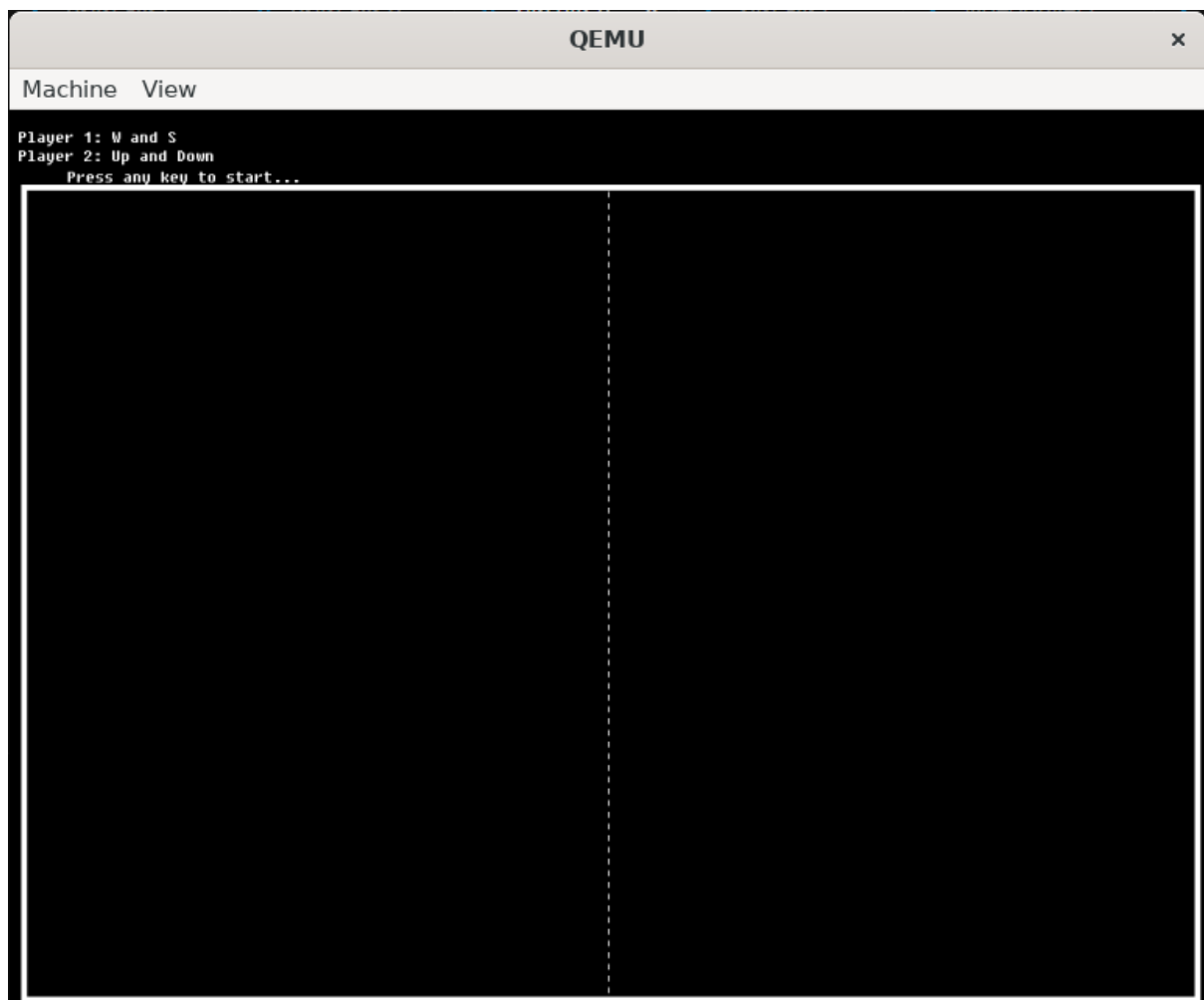


Pong:

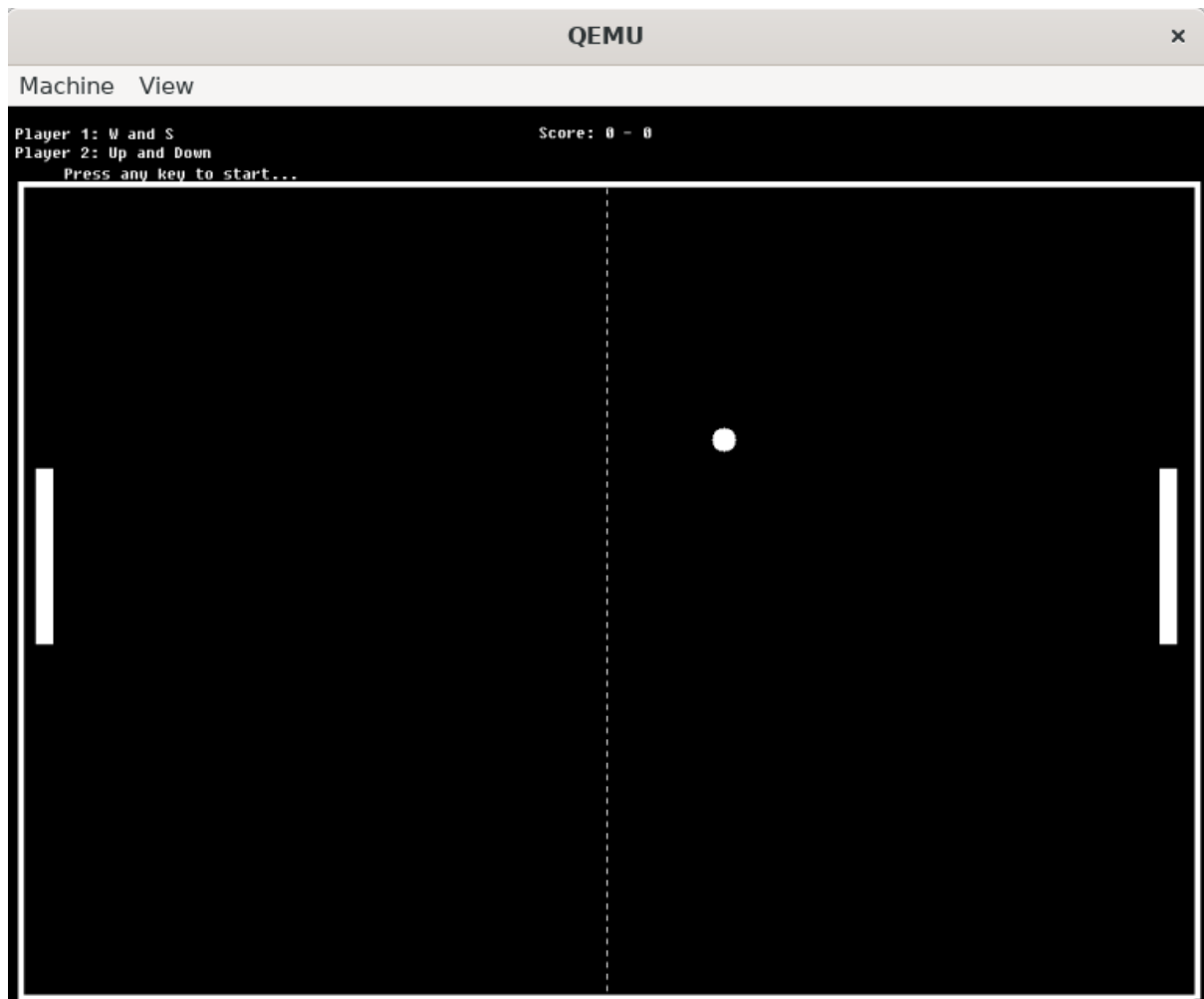
El “pong” es un juego que consiste de 2 jugadores y una pelota. El objetivo de un jugador es realizar goles al otro para ganarle la partida. Una partida consiste en una pelota, la cual va rebotando en los jugadores y en los extremos inferiores y superiores de la “cancha”. Si un jugador no consigue “atajar” la pelota y hacer que rebote en sí mismo y esta, toca el extremo de la cancha que se encuentra detrás del jugador, se realiza un gol. El jugador 1 (el cual se encuentra en la izquierda de la pantalla) se mueve para arriba con la tecla “*W*” y para abajo con la tecla “*S*” mientras que el jugador 2 (el cual se encuentra a la derecha de la pantalla) se mueve para arriba con la tecla de la **flecha hacia arriba** y para abajo con la tecla de la **flecha hacia abajo**.

En el caso de querer pausar el juego presionar la tecla “*p*” y para volver a ponerle play al juego volver a presionarla. En el caso de querer terminar una partida antes de que la misma se termine presionar la tecla “*Esc*” y será llevado a la terminal.

Al entrar al pong, aparece la pantalla con el juego pero vacía, en el caso de querer comenzar a jugar presionar cualquier tecla como indicado en la siguiente pantalla:



Una vez presionada la tecla aparecen los jugadores y la pelota.



Como se observa en la imagen, en la parte superior aparece el puntaje de la partida. Una vez que algún jugador llegue a 3 puntos, lo cual significa que ganó la partida, aparece la siguiente pantalla donde el usuario puede elegir volver a jugar (presionando la tecla “r”) o ir hacia la terminal (presionando **cualquier tecla** menos la “r”).

```
Player 1 wins!  
Press R to play again or any key to continue...
```