



# **Arquitectura de Computadoras**

## **Manual de Usuario TPE**

Integrantes:

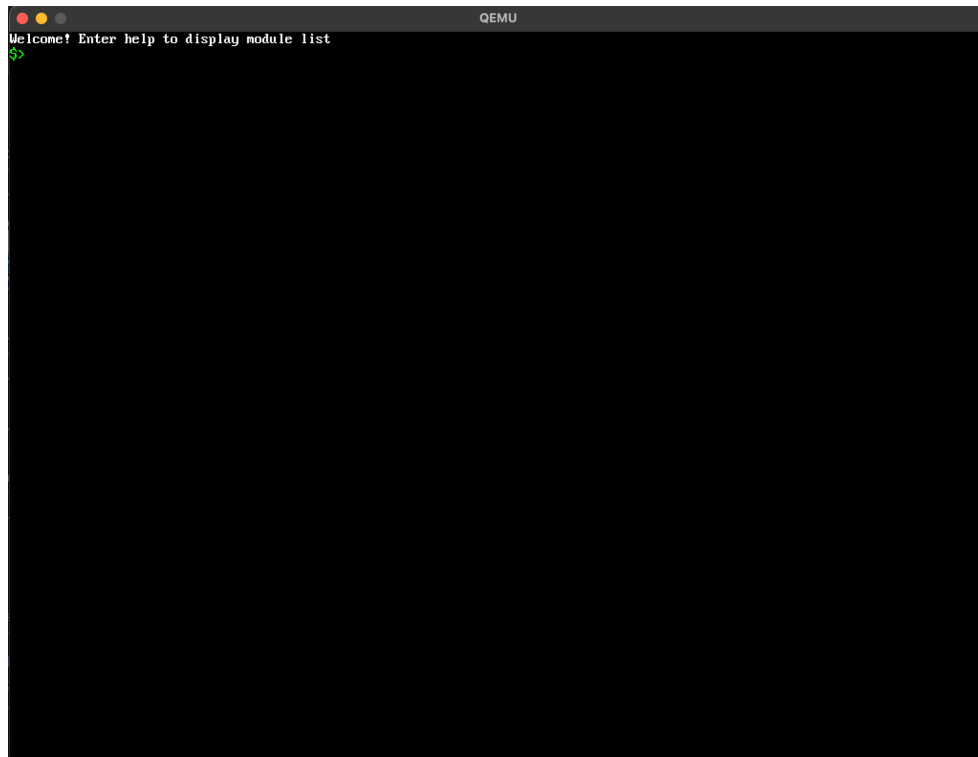
Axel Castro Benza - 62358 - [acastrobenza@itba.edu.ar](mailto:acastrobenza@itba.edu.ar)

Rocio D'Intino - 62341 - [rdintino@itba.edu.ar](mailto:rdintino@itba.edu.ar)

Ian Franco Tognetti - 61215 - [itognetti@itba.edu.ar](mailto:itognetti@itba.edu.ar)

## Acceso al sistema

Ejecutando el archivo run.sh. Le aparecerá la siguiente pantalla.

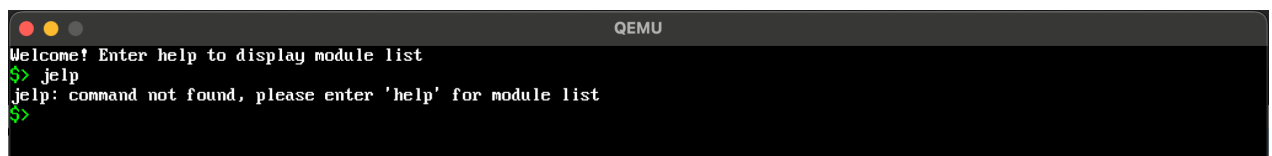


*Fig 1. Pantalla de bienvenida*

En este intérprete de comandos se verá un saludo de bienvenida seguido de una línea de comando. A partir de este momento usted debe escribir los comandos que desea, siempre y cuando estén permitidos.

## Comandos

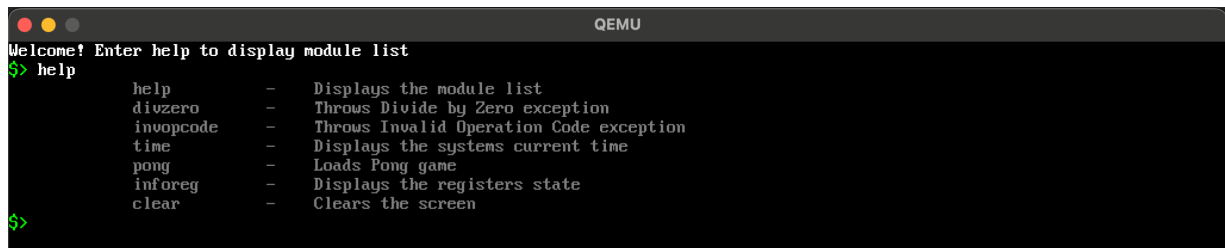
Existen siete comandos que el intérprete dará como válidos. Los siguientes son: *help*, *divzero*, *invopcode*, *time*, *pong*, *inforeg* y *clear*. En caso de escribir mal un comando o redactar uno inválido se mostrará la siguiente imagen.



*Fig 2. Pantalla tras ejecutar un comando no válido*

## Help

Este comando muestra los otros válidos junto con una breve descripción de los mismos. Al escribirlo se le presentará la siguiente pantalla.

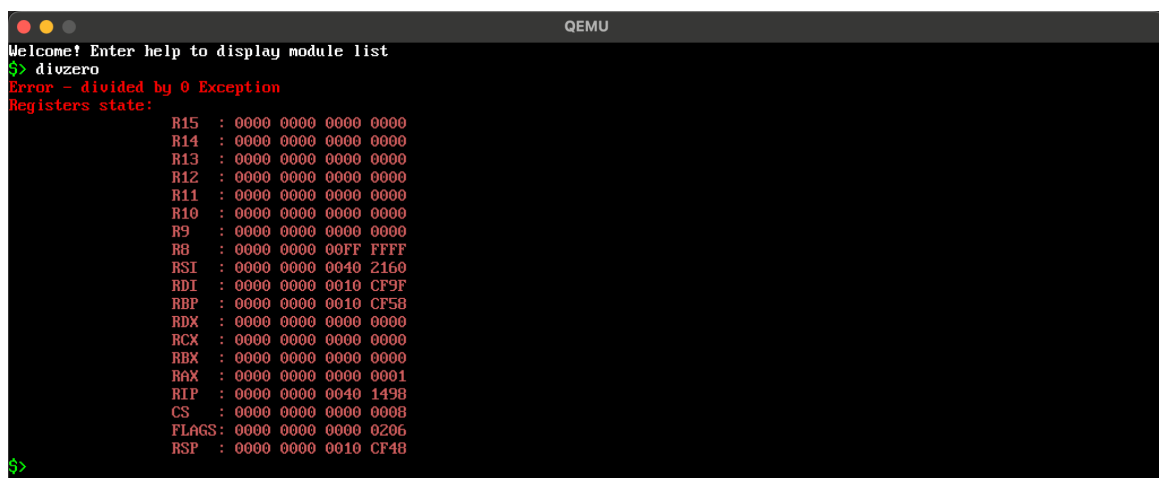


```
QEMU
Welcome! Enter help to display module list
$> help
      help      -   Displays the module list
      divzero    -   Throws Divide by Zero exception
      invopcode  -   Throws Invalid Operation Code exception
      time       -   Displays the systems current time
      pong       -   Loads Pong game
      inforeg    -   Displays the registers state
      clear      -   Clears the screen
$>
```

*Fig 3. Pantalla tras ejecutar el comando help*

## Divzero

Imprime en pantalla los valores de los registros a la hora de invocar la excepción “Divide By Zero”. Se le exhibirá una imagen similar.

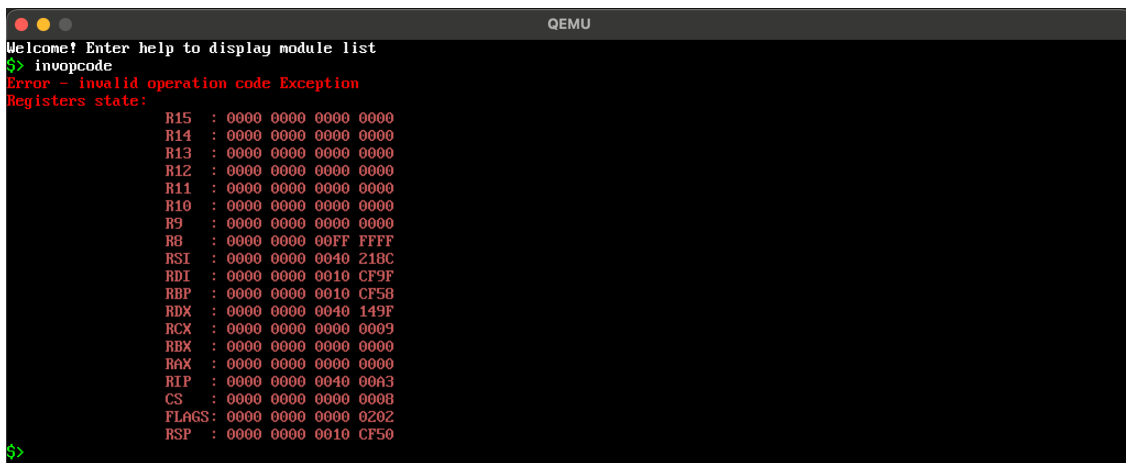


```
QEMU
Welcome! Enter help to display module list
$> divzero
Error - divided by 0 Exception
Registers state:
      R15 : 0000 0000 0000 0000
      R14 : 0000 0000 0000 0000
      R13 : 0000 0000 0000 0000
      R12 : 0000 0000 0000 0000
      R11 : 0000 0000 0000 0000
      R10 : 0000 0000 0000 0000
      R9  : 0000 0000 0000 0000
      R8  : 0000 0000 00FF FFFF
      RSI : 0000 0000 0040 2160
      RDI : 0000 0000 0010 CF9F
      RBP : 0000 0000 0010 CF58
      RDX : 0000 0000 0000 0000
      RCX : 0000 0000 0000 0000
      RBX : 0000 0000 0000 0000
      RAX : 0000 0000 0000 0001
      RIP : 0000 0000 0040 1498
      CS  : 0000 0000 0000 0008
      FLAGS: 0000 0000 0000 0206
      RSP : 0000 0000 0010 CF48
$>
```

*Fig 4. Pantalla tras ejecutar el comando divzero*

## Invopcode

Plasma en pantalla los valores de los registros al momento de invocar la excepción “Invalid Operation Code”. Se le mostrará la siguiente imagen.



```
QEMU
Welcome! Enter help to display module list
$> invopcode
Error - invalid operation code Exception
Registers state:
R15 : 0000 0000 0000 0000
R14 : 0000 0000 0000 0000
R13 : 0000 0000 0000 0000
R12 : 0000 0000 0000 0000
R11 : 0000 0000 0000 0000
R10 : 0000 0000 0000 0000
R9 : 0000 0000 0000 0000
R8 : 0000 0000 00FF FFFF
RSI : 0000 0000 0040 218C
RDI : 0000 0000 0010 CF9F
RBP : 0000 0000 0010 CF58
RDX : 0000 0000 0040 149F
RCX : 0000 0000 0000 0009
RBX : 0000 0000 0000 0000
RAX : 0000 0000 0000 0000
RIP : 0000 0000 0040 00A3
CS : 0000 0000 0000 0008
FLAGS: 0000 0000 0000 0202
RSP : 0000 0000 0010 CF50
$>
```

*Fig 5. Pantalla tras ejecutar el comando invopcode*

## Time

Este comando muestra la fecha y hora actual en formato GMT - 3. Se desplegará la siguiente pantalla.

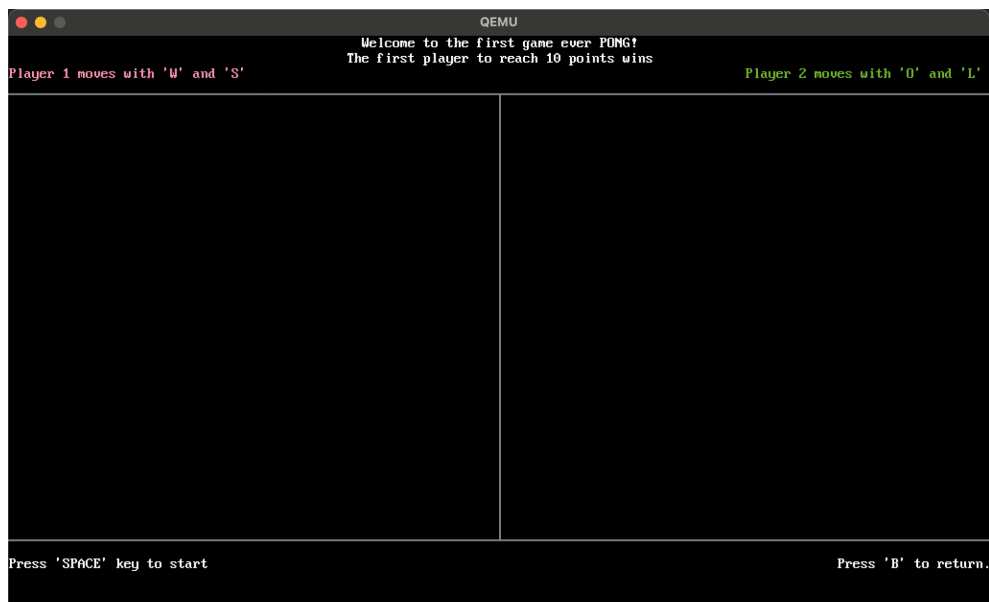


```
QEMU
Welcome! Enter help to display module list
$> time
Local time & date: 17:33:43 04/06/23
$>
```

*Fig 6. Pantalla tras ejecutar el comando time*

## Pong

Al ejecutar el comando, ingresará al juego llamado pong y se le mostrará la siguiente imagen.



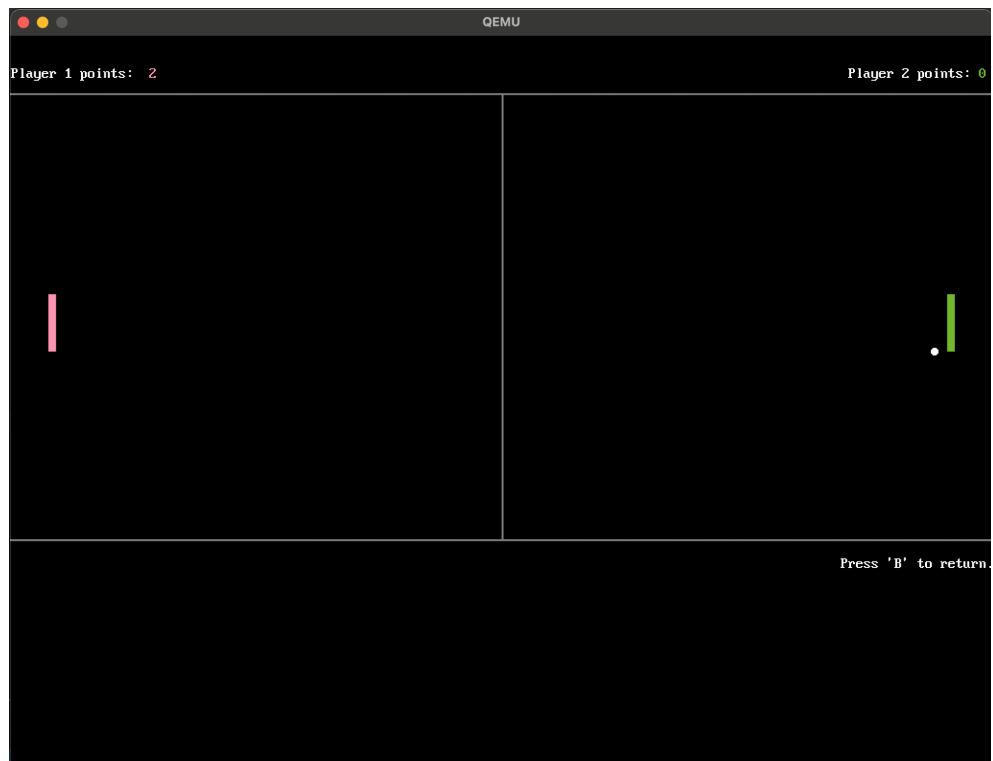
```
QEMU
Welcome to the first game ever PONG!
The first player to reach 10 points wins

Player 1 moves with 'W' and 'S'
Player 2 moves with 'D' and 'L'

Press 'SPACE' key to start
Press 'B' to return.
```

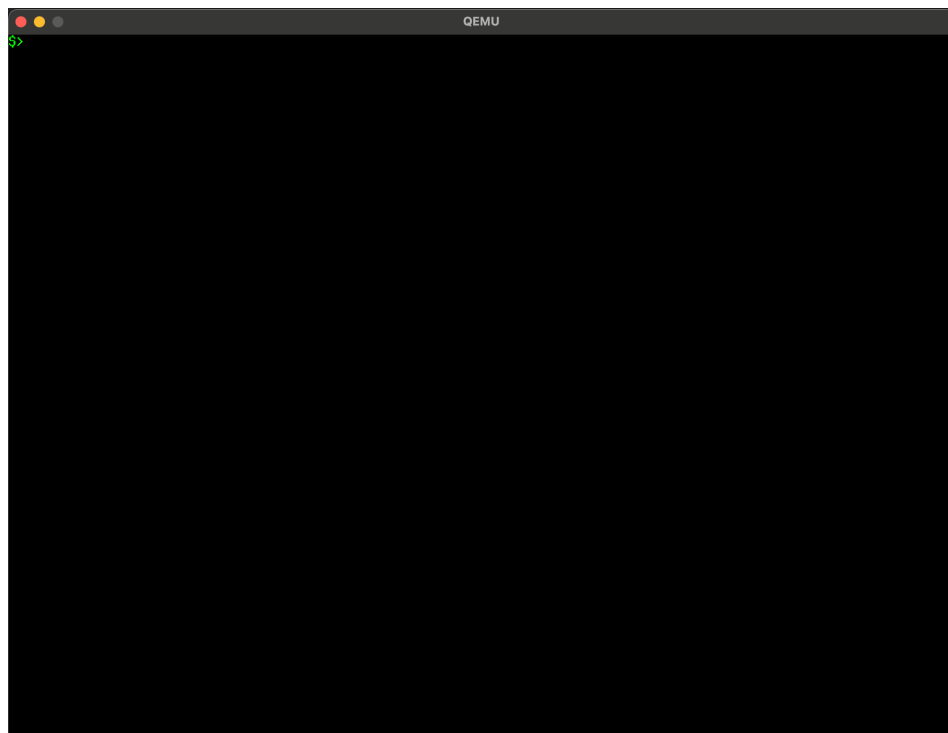
*Fig 7. Pantalla tras ejecutar el comando pong*

Esta es la pantalla principal de juego, en caso de querer jugarlo deberá presionar la barra espaciadora (SPACE). Disfrute!



*Fig 8. Pantalla tras apretar la barra espaciadora*

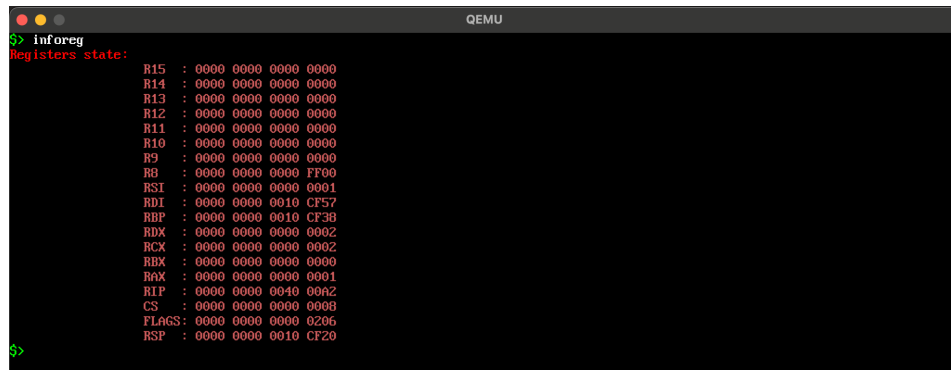
Si desea volver al intérprete de comandos, tendrá que presionar la tecla B.



*Fig 9. Pantalla tras tocar la tecla B*

## InfoREG

Imprime en la pantalla los valores de los registros al momento de apretar las teclas CTRL + R. Se le presentará la siguiente imagen.

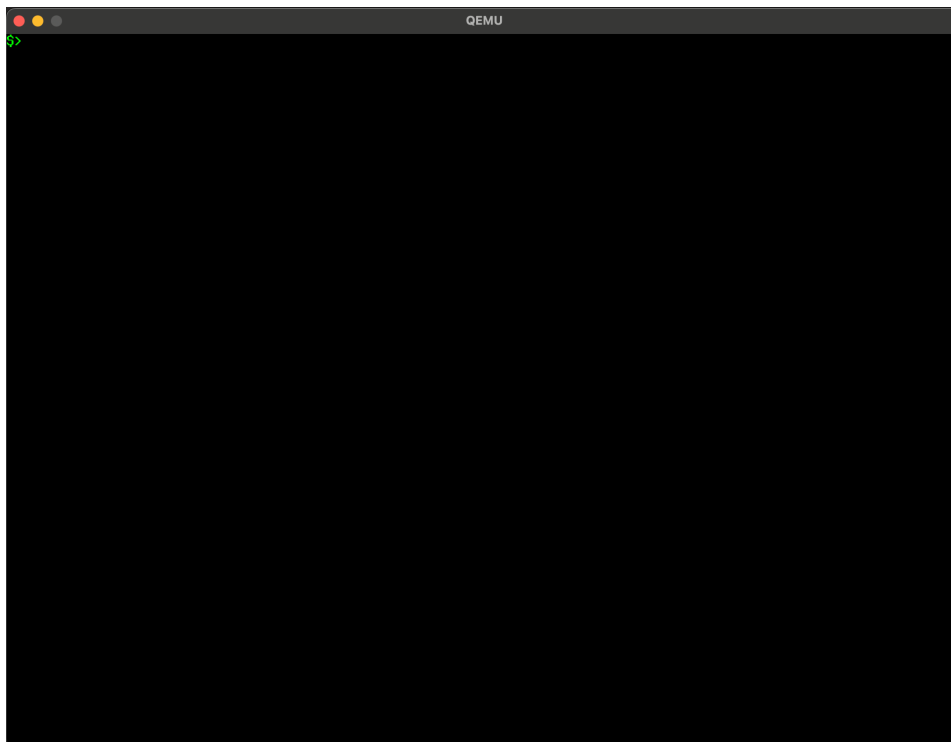


```
$> infoREG
Registers state:
R15 : 0000 0000 0000 0000
R14 : 0000 0000 0000 0000
R13 : 0000 0000 0000 0000
R12 : 0000 0000 0000 0000
R11 : 0000 0000 0000 0000
R10 : 0000 0000 0000 0000
R9 : 0000 0000 0000 0000
R8 : 0000 0000 0000 FF00
RSI : 0000 0000 0000 0001
RDI : 0000 0000 0010 CF57
RBP : 0000 0000 0010 CF3B
RDX : 0000 0000 0000 0002
RCX : 0000 0000 0000 0002
RBX : 0000 0000 0000 0000
RAX : 0000 0000 0000 0001
RIP : 0000 0000 0040 00A2
CS : 0000 0000 0000 0008
FLAGS: 0000 0000 0000 0206
RSP : 0000 0000 0010 CF20
$>
```

*Fig 10. Pantalla tras ejecutar el comando infoREG después de jugar al pong y apretar las teclas CTRL + R.*

## Clear

Elimina todas las líneas de comando previamente escritas. Restableciendo la posición a escribir a la default.



*Fig 11. Pantalla tras ejecutar el comando clear*