

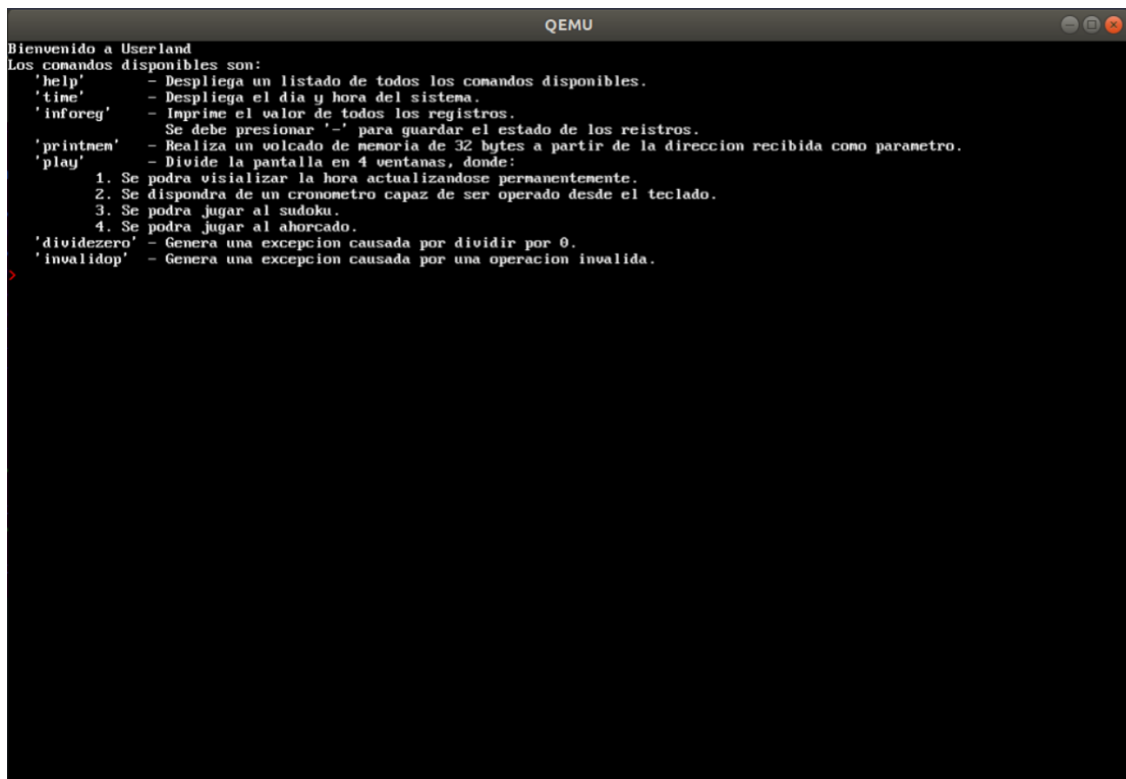
Manual de Usuario - Pure64

Ferreiro, Lucas Agustín (61595)
lferreiro@itba.edu.ar

Di Toro, Camila (62576)
cditoro@itba.edu.ar

Chayer, Iván (61360)
ichayer@itba.edu.ar

Grupo 15 - Segundo cuatrimestre 2021



```
QEMU
Bienvenido a Userland
Los comandos disponibles son:
'help'      - Despliega un listado de todos los comandos disponibles.
'time'      - Despliega el día y hora del sistema.
'inforeg'   - Imprime el valor de todos los registros.
             Se debe presionar '-' para guardar el estado de los registros.
'printmem'  - Realiza un volcado de memoria de 32 bytes a partir de la dirección recibida como parámetro.
'play'      - Divide la pantalla en 4 ventanas, donde:
               1. Se podrá visualizar la hora actualizándose permanentemente.
               2. Se dispondrá de un cronómetro capaz de ser operado desde el teclado.
               3. Se podrá jugar al sudoku.
               4. Se podrá jugar al ahorcado.
'dividezero' - Genera una excepción causada por dividir por 0.
'invalidop' - Genera una excepción causada por una operación inválida.
```

COMANDOS DEL USER-SPACE

Los comandos a los que puede acceder el usuario se escriben en la *Shell* una vez iniciado *Qemu*. Para escribir una letra en mayúscula se tiene que mantener apretada la tecla *Shift*. Los comandos disponibles son:

(1) help

Muestra en pantalla un listado con todos los comandos disponibles del *User Space*.

```
1 >>help
  Los comandos disponibles son:
3 - help - Despliega un listado de todos los comandos disponibles.
  - time - Despliega el dia y hora del sistema.
5 - inforeg - Imprime el valor de todos los registros.
               Se debe presionar '-' para guardar el estado de los
7               registros.
  - printmem - Realiza un volcado de memoria de 32 bytes a partir de la
9               direccion recibida como parametro.
  - play - Divide la pantalla en 4 ventanas, donde:
11     1. Se podra visualizar la hora actualizandose permanentemente.
     2. Se dispondra de un cronometro capaz de ser operado desde el
13     teclado.
     3. Se podra jugar al sudoku.
15     4. Se podra jugar al ahorcado.
  - dividezero - Genera una excepcion causada por dividir por 0.
17 - invalidop - Genera una excepcion causada por una operacion invalida.
```

(2) time

Muestra en pantalla la fecha y hora del sistema.

```
1 >>time
  - Hora: 19:55:39
3  - Fecha: 05/11/2021
```

(3) inforeg

Imprime en pantalla el estado de todos los registros. El usuario debe presionar la tecla "-" para guardar el valor de todos los registros en el instante que la tecla fue presionada. Al utilizar el comando *inforeg* se imprimen dichos registros guardados.

```
1 >>inforeg
  Debes presionar "-" para guardar el estado de los registros en un
3  momento especifico.
  >>inforeg
5  - RAX = 0x0000000000000000
  - R15 = 0x0000000000000000
7  - R14 = 0x0000000000000000
  - R13 = 0x0000000000000000
9  - R12 = 0x0000000000000000
  - R11 = 0x0000000000000000
11 - R10 = 0x0000000000000000
  - R9 = 0x0000000000000000
13 - R8 = 0x0000000000000000
  - RSI = 0x000000000010EF13
```

```

15 - RDI = 0x0000000000000000
    - RBP = 0x0000000000010EEF8
17 - RDX = 0x0000000000000001
    - RCX = 0x0000000000010EFB0
19 - RBX = 0x0000000000000000
    - RSP = 0x0000000000010EEC8

```

(4) printmem

Realiza un volcado de memoria de 32 bytes a partir de la dirección de memoria de 64 bits recibida como parámetro. Por ejemplo, ingresando la dirección de memoria 55:

```

>>printmem
2  Ingrese una direccion de memoria en formato hexadecimal: 55
    - 55 = 8E
4  - 56 = 0
    - 57 = 0
6  - 58 = 0
    - 59 = 0
8  - 5A = 0
    - 5B = 0
10 - 5C = 0
    - 5D = 0
12 - 5E = 0
    - 5F = 0
14 - 60 = FB
    - 61 = 24
16 - 62 = 8
    - 63 = 0
18 - 64 = 0
    - 65 = 8E
20 - 66 = 10
    - 67 = 0
22 - 68 = 0
    - 69 = 0
24 - 6A = 0
    - 6B = 0
26 - 6C = 0
    - 6D = 0
28 - 6E = 0
    - 6F = 0
30 - 70 = 97
    - 71 = 90
32 - 72 = 8
    - 73 = 0
34 - 74 = 0

```

Con este comando pueden ocurrir tres tipos de errores, el primero ocurre al ingresar un formato hexadecimal inválido:

```

>>printmem
2  Ingrese una direccion de memoria en formato hexadecimal: y5
    Formato hexa invalido

```

El segundo error ocurre cuando el valor ingresado sobrepasa 512MB, que es la cantidad de memoria accesible/disponible de *Qemu*:

```
1 >>printmem
   Ingrese una direccion de memoria en formato hexadecimal: 500000000
3 Acceso a memoria invalido
```

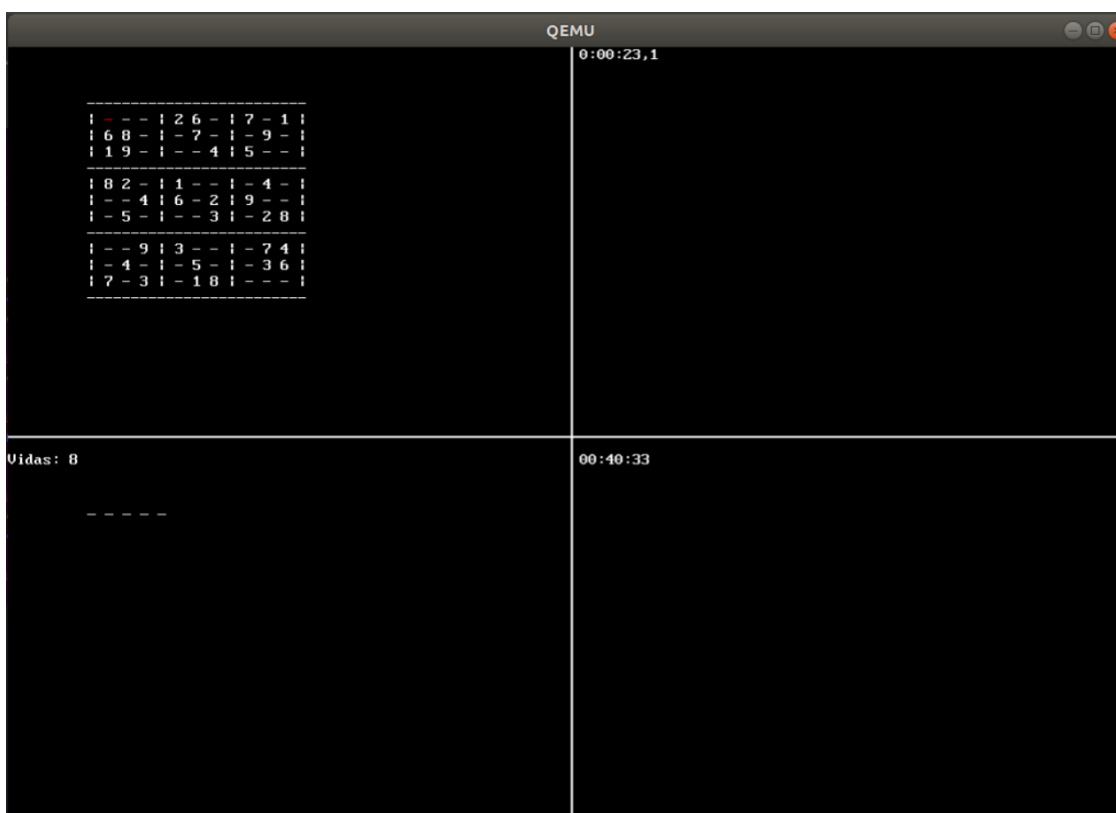
El tercer error ocurre cuando el valor ingresado supera más de 64 bits (16 digitos hexa):

```
1 >>printmem
   Ingrese una direccion de memoria en formato hexadecimal: 50000000000000
3 0000
   La direccion ingresada no puede tener mas de 64 bits (16 digitos hexa)
```

(5) play

Divide la pantalla en 4 ventanas. En una se muestra la hora y en otra un cronómetro que se actualiza en forma constante. En las otras dos ventanas, se podrá jugar al sudoku y al ahorcado.

```
>>play
```



- Con las flechas del teclado uno puede moverse por el sudoku y presionando cualquier tecla numérica, se ingresa el número deseado en el lugar donde se encuentre el cursor. En el caso de que en dicha posición ya haya un número, este es reemplazado (excepto aquellos números que ya se encontraban fijos).
- Con la tecla + se pausa/reanuda el cronómetro, y con la tecla 0 se reinicia este último y queda a la espera de su reanudación.
- Con cualquier letra alfabética del teclado (tanto en mayúscula como en minúscula) se podrá jugar al ahorcado.
- Con la tecla Espacio se abandona el modo *play* y se vuelve a la *Shell*.

(6) dividezero

Genera una excepción por intentar dividir por cero e imprime en pantalla el valor de los registros al generarse la excepción.

```
1      >>dividezero
      - RAX = 0000000000000000
3      - RBX = 0000000000000000
      - RCX = 0000000000000000
5      - RDX = 0000000000000006
      - RBP = 000000000010FDA0
7      - RDI = 000000000000005F
      - RSI = 0000000000000080
9      - R8 = 0000000000000001
      - R9 = 0000000000000000
11     - R10 = 0000000000000000
      - R11 = 0000000000000000
13     - R12 = 0000000000000000
      - R13 = 0000000000000000
15     - R14 = 0000000000000000
      - R15 = 0000000000000000
17     - RIP = 0000000000101BA0
      - CS = 0000001000000008
19     - RFLAGS = 000000000103197
      - RSP = 0000000000101E57
21     - SS = 0000000A00000000
      Zero Division Exception
23     Presione enter para continuar
```

(7) invalidop

Genera una excepción por una operación inválida e imprime en pantalla el valor de los registros al generarse la excepción.

```
1      >>invalidop
      - RAX = 0000000000000000
3      - RBX = 0000000000000000
      - RCX = 0000000000000000
5      - RDX = 0000000000000003
      - RBP = 000000000010FDA0
7      - RDI = 000000000000002F
      - RSI = 0000000000000080
9      - R8 = 0000000000000001
      - R9 = 0000000000000000
11     - R10 = 0000000000000000
      - R11 = 0000000000000000
13     - R12 = 0000000000000000
      - R13 = 0000000000000000
15     - R14 = 0000000000000000
      - R15 = 0000000000000000
17     - RIP = 0000000000101BA0
      - CS = 0000001000000008
19     - RFLAGS = 0000000001031AF
      - RSP = 0000000000101E57
21     - SS = 0000000A00000000
```

```
Invalid Operation exception  
Presione enter para continuar
```