

Trabajo práctico especial

Arquitectura de las computadoras

Juan Pablo Dantur - 54623
Juan Martín Pascale - 55027

Junio de 2015

Objetivo

Realizar un sistema de 64 bits que administre los recursos de hardware de una computadora y muestre características del modo protegido de Intel.

Estructura básica

Se programó el sistema respetando la separación por capas entre el Kernel y el User Space. La comunicación se realiza únicamente mediante system calls. Se programó la interrupción de software INT 80h y las siguientes llamadas a sistema:

- READ
- WRITE
- TIME_READ
- TIME_WRITE
- SCREEN_SAVER_SET

A continuación se detalla el estado registros para realizar cada llamada:

Syscall	RAX	RBX	RCX	RDX
READ	3	int: tamaño del string a leer.	-	-
WRITE	4	int: tamaño del string a imprimir.	-	-
TIME_READ	2	Int: Identificador de atributo del tiempo a leer.	-	-
TIME_WRITE	5	Int: Identificador de atributo del tiempo a escribir.	-	Int: Valor del atributo a escribir en hexadecimal.
SCREEN_SAVER_SET	6	-	-	Int: Tiempo especificado para la activación del protector de pantalla. (En seg/18.18182)

Kernel-land

Kernel

El kernel está directamente en contacto con el hardware y atiende los pedidos provenientes del usuario a través de la INT 80h. Al bootear el sistema, el kernel borra la pantalla, carga las interrupciones en la IDT asociando cada una con su rutina de atención de interrupción, escribe la máscara del PIC (Master y Slave) de manera tal que se permiten las interrupciones del Timer Tick y del Keyboard, habilita las interrupciones enmascarables realizando una llamada a una función de assembler y carga el módulo de usuario. Después de eso, queda en espera para atender interrupciones de software y de hardware.

Keyboard Driver

Entre las funciones que ofrece el driver de teclado se encuentran:

keyboard_buffer_write: es invocada por la rutina de atención de la interrupción de teclado. La misma se encarga de obtener el código de la tecla presionada (o soltada) de la dirección 0x60 de E/S y escribir en un buffer el carácter. Para esto, se tiene en cuenta que el carácter se puede obtener de uno de los dos vectores estáticos que contienen los caracteres, según si se encuentra presionada la tecla Shift o no. Además, cada interrupción generada llama a una función del driver de video para imprimir la tecla en pantalla con la finalidad de que el usuario sepa que tecla fue presionada.

keyboard_buffer_read: queda en espera hasta que el índice de escritura del buffer sea mayor al de lectura. Al darse esa situación, devuelve el proximo caracter a leer.

El teclado cuenta con un buffer que se implementa de manera circular, es decir, se tiene un vector de 256 caracteres y se insertan/obtienen los valores utilizando dos índices (uno para inserciones y otro para lecturas), los cuales se incrementan en 1 siempre luego de su respectiva operación. En el vector, se accede a la posición del índice (Aplicando módulo 256), ya que de esta forma el índice puede crecer libremente, y el vector nunca debe ser agrandado.

Video Driver

Es el encargado de administrar la memoria asignada a video. Lleva cuenta y manejo del puntero actual en donde se debe escribir. Siempre que se desee escribir datos en pantalla, se realizan llamadas a funciones de este controlador y estas deciden la posición de memoria en la que se los coloca.

Al espacio continuo asignado a video se lo trata como una matriz de 80 filas por 25 columnas y se realizan los cálculos correspondientes a través de las funciones. Por ejemplo, para generar una nueva línea, se completa con ceros y se avanza el puntero actual de video hasta llegar a una posición correspondiente con un inicio de fila, es decir, la primera posición.

Se implementó la funcionalidad del backspace en video que tiene como objetivo eliminar un carácter de la pantalla. Los únicos caracteres que pueden ser borrados son aquellos provenientes de interrupciones generadas por el teclado. Para esto, el driver ofrece funciones llamadas directamente por el driver de teclado que internamente implementan un contador. Cada vez que se genera una interrupción de teclado (de una tecla que no sea el backspace) y se escribe en pantalla, se suma uno al contador. Al generarse una interrupción cuyo código de tecla sea el backspace, el driver borra un carácter y actualiza el puntero a la nueva posición de memoria (2 bytes menos), siempre y cuando el contador no sea 0. Esto evita que se borren caracteres que no fueron escritos mediante interrupciones del teclado (Ejemplo: El prompt de la consola).

Además, se cuentan con funciones de manejo de screen saver (Protector de pantalla) y se ofrecen funciones para el control del mismo. Con cada interrupción del Timer Tick se controla la cuenta de inactividad para que el mismo se dispare y con cada interrupción del teclado se resetea. Cuando este se dispara, se copia la porción de memoria asignada a video en un vector auxiliar y se muestra el protector. Ante una interrupción de teclado, se borra la pantalla y se procede a restablecer la pantalla a su estado previo al protector de pantalla.

Manejo RTC

El kernel cuenta con funciones que interactúan directamente con el hardware para obtener o modificar la hora del sistema. Cada atributo del tiempo se modifica por separado. En la dirección 0x70 de E/S se inserta el tipo de dato que se va a leer o escribir (0x00 para los segundos, 0x02 para los minutos, etc) y en la dirección 0x71 de E/S se lee (o se escribe) el dato en cuestión. El usuario puede acceder a esta funcionalidad mediante llamados a sistema.

User-land

Consola

Al inicio del programa, la consola le pregunta al usuario cual es su nombre, y luego entra en un ciclo infinito en el cual escribe como prompt el nombre del usuario (utilizando print de la librería standard), y espera que el usuario ingrese un comando (mediante scan), el cual finaliza al detectar '\n' o cuando se llena el buffer. Mediante una llamada al sistema se obtiene una copia del buffer de teclado (el espacio

utilizado es creado por el usuario y escrito por el Kernel). Luego, se procede a parsear y validar la información obtenida en el buffer. Si el comando inválido la consola imprimirá “Comando invalido”, mientras que si es válido se procederá a ejecutar el mismo.

Funciones de la librería standard de C

El sistema cuenta con las funciones `getchar`, `putchar`, `scan`, `print`, `printint` y `printhex`. Las funciones `putchar` y `getchar` se encargan de realizar los llamados a sistema `write` y `read`, respectivamente, para imprimir en pantalla u obtener del buffer un carácter, según el caso. `Scan` tiene una funcionalidad similar a `getchar`, pero para una cantidad dada de caracteres. `Print` recibe un puntero a un string en memoria finalizado en ‘\0’ y se encarga de imprimirlo en pantalla (Los caracteres impresos mediante llamados a sistema no pueden ser borrados). `Printint` se encarga de pasar un número entero a string e imprime ese valor en pantalla. `printhex` tiene una funcionalidad similar con números en hexadecimal, anteponiendo un “print 0x” (Por ejemplo, recibe 0x23 e imprime “0x23”).

Según si se va a imprimir un carácter, o a leer uno, se especifica un cierto parámetro en la `system_call`, el cual definirá si se llama a la función `Read` o `Write` del sistema.

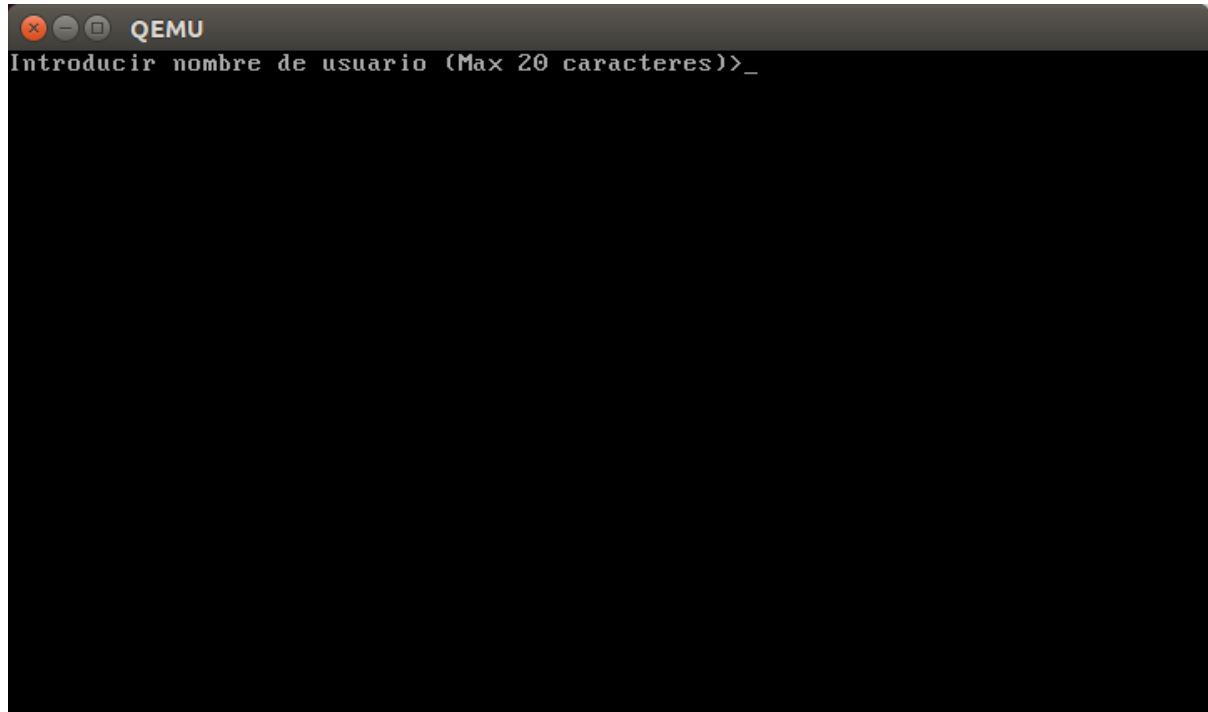
Comandos

Los comandos disponibles en la consola son los siguientes:

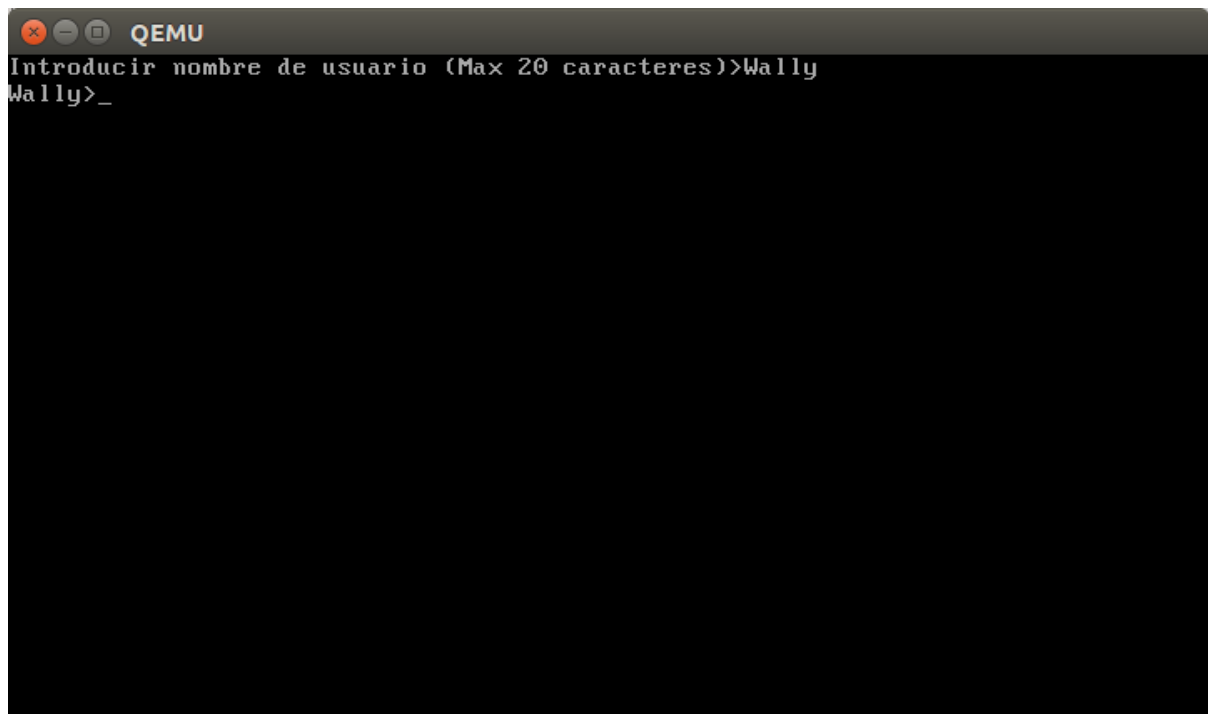
- *help*: Muestra todos los comandos
- *echo [arg]*: imprime el argumento
- *set_time HH:MM:SS* setea la hora en el formato dado.
- *set_hour [x]*: setea la hora en x
- *set_min [x]*: setea los min en x
- *set_sec [x]*: setea los segundos en x
- *set_screensaver [x]*: setea el tiempo de espera hasta que el screensaver se active en x segundos
- *test*: ejecuta una prueba de `printint`, `print` y `printhex`
- *time*: Imprime la hora en formato HH:MM:SS

Manual de usuario

Al ingresar al sistema, este solicitará ingresar el nombre de usuario:

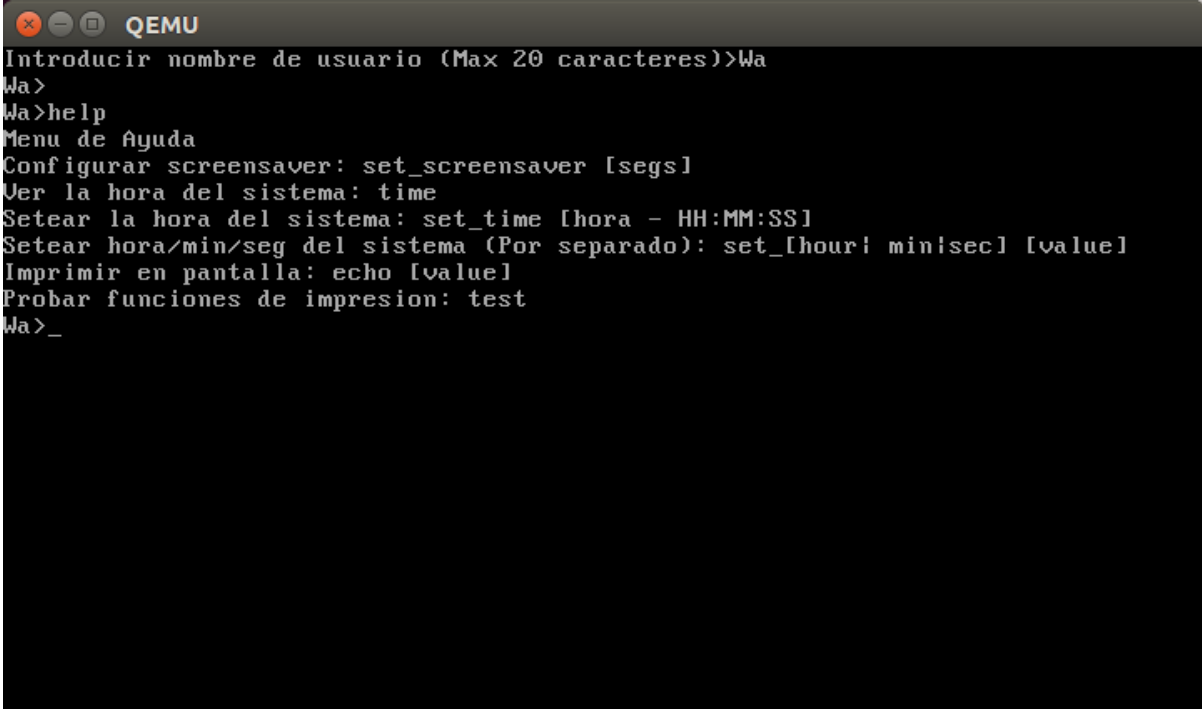


Al ingresarlo y presionar enter se activará un prompt con ese nombre de usuario.



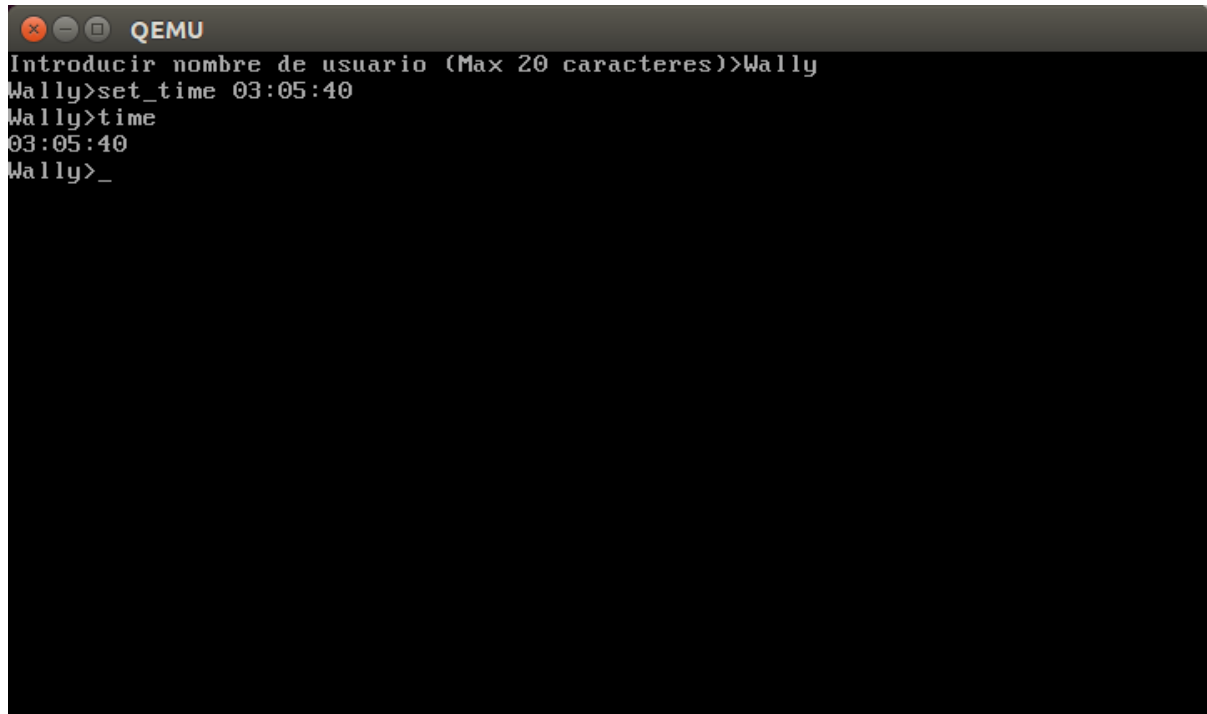
Help

Al tipear *help* en la consola se muestra una lista de comandos disponibles:

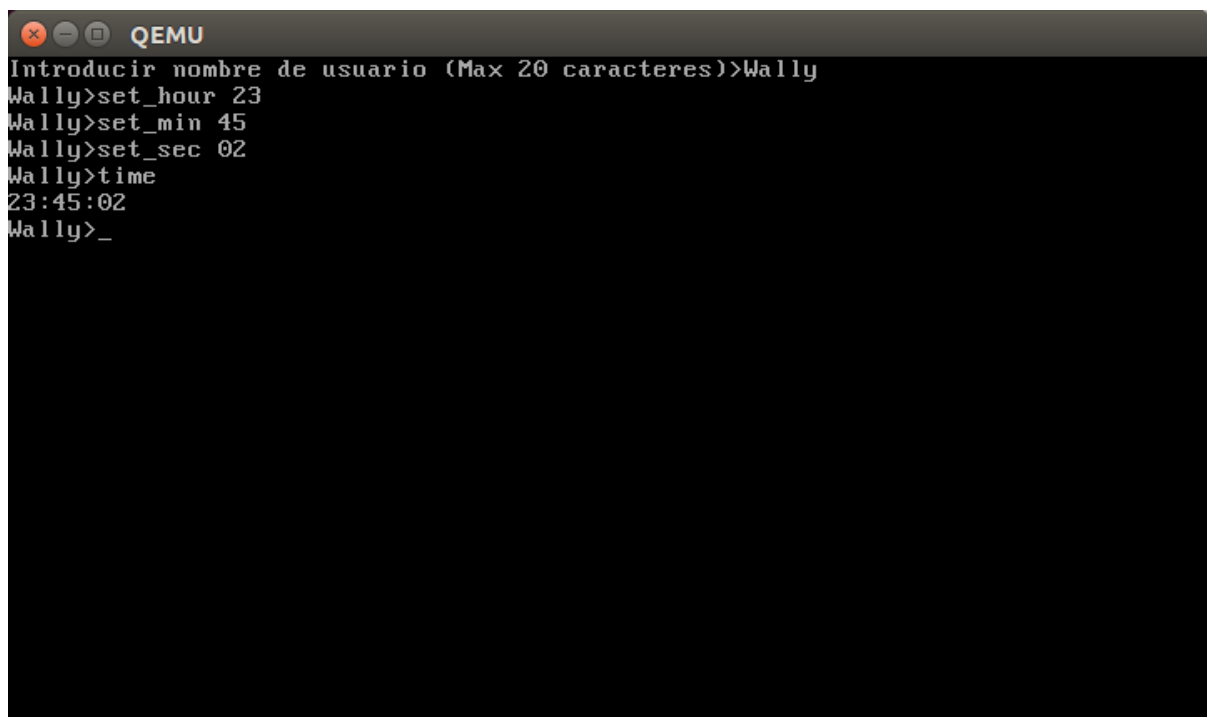
A screenshot of a QEMU terminal window. The window has a title bar with standard Linux window controls (close, maximize, minimize) and the text 'QEMU'. The terminal content shows a user prompt 'Introducir nombre de usuario (Max 20 caracteres)>' followed by the input 'Wa'. The prompt changes to 'Wa>'. The user enters 'help', and the terminal displays a 'Menu de Ayuda' (Help Menu) with the following commands: 'Configurar screensaver: set_screensaver [secs]', 'Ver la hora del sistema: time', 'Setear la hora del sistema: set_time [hora - HH:MM:SS]', 'Setear hora/min/seg del sistema (Por separado): set_[hour| min|sec] [value]', 'Imprimir en pantalla: echo [value]', and 'Probar funciones de impresion: test'. The prompt returns to 'Wa>'.

Hora del sistema

Es posible cambiar la hora del sistema de dos maneras, ya sea cambiando la hora completa o uno de sus atributos (hora, minutos, segundos). La primer manera es mediante el comando `set_time HH:MM:SS` y la segunda mediante `set_[hour | min | sec] [value]`. La hora actual del sistema se puede obtener mediante el comando `time`. Todo esto se detalla en las siguientes imágenes:



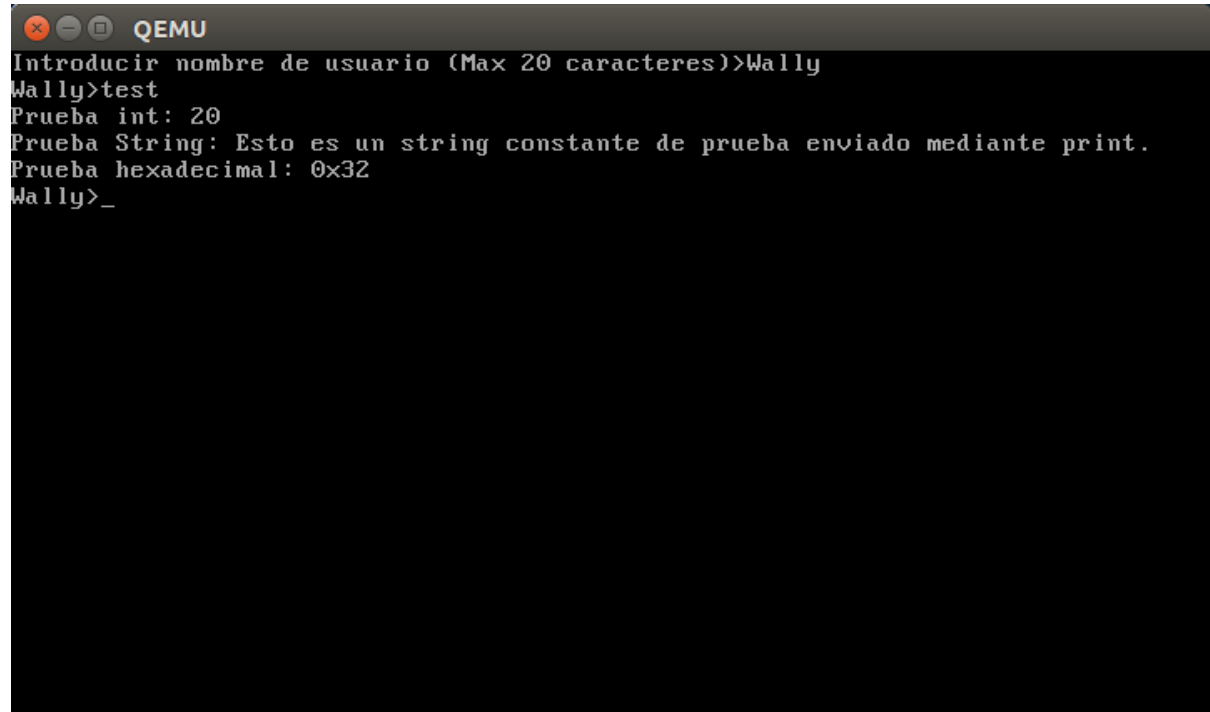
```
QEMU
Introducir nombre de usuario (Max 20 caracteres)>Wally
Wally>set_time 03:05:40
Wally>time
03:05:40
Wally>_
```



```
QEMU
Introducir nombre de usuario (Max 20 caracteres)>Wally
Wally>set_hour 23
Wally>set_min 45
Wally>set_sec 02
Wally>time
23:45:02
Wally>_
```


Test

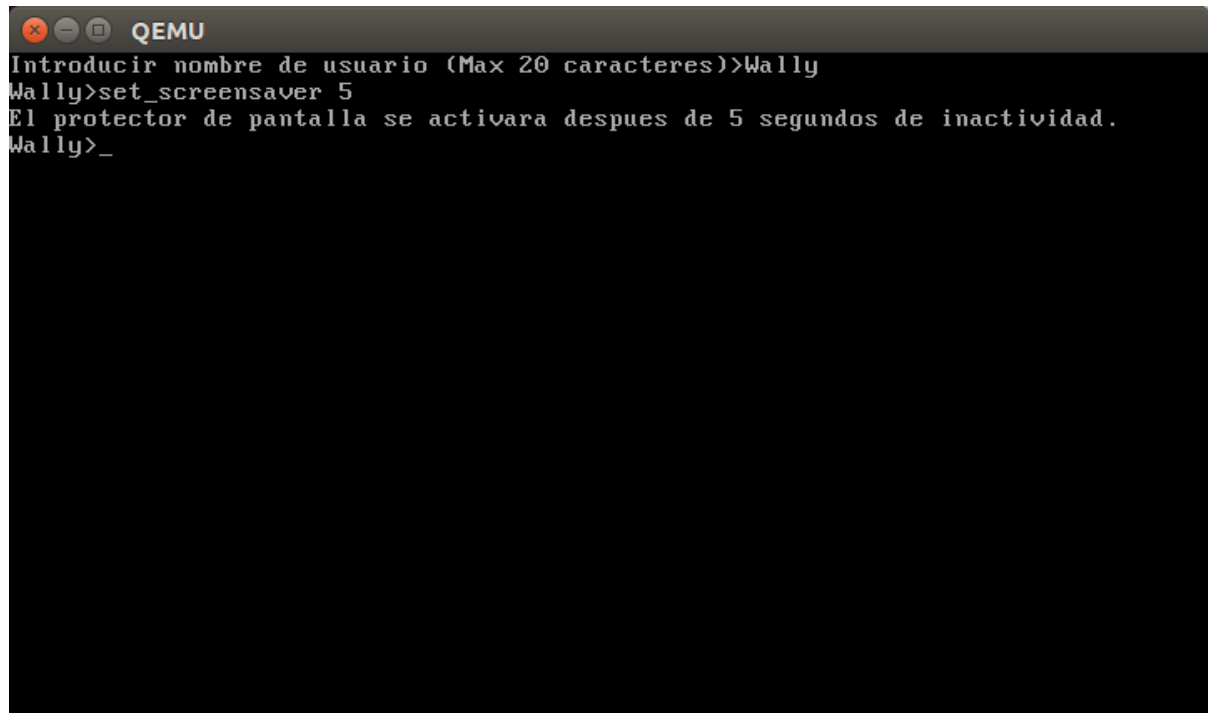
El comando *test* permite mostrar el funcionamiento de las funciones `print`, `printint` y `printhex`:



```
QEMU
Introducir nombre de usuario (Max 20 caracteres)>Wally
Wally>test
Prueba int: 20
Prueba String: Esto es un string constante de prueba enviado mediante print.
Prueba hexadecimal: 0x32
Wally>_
```

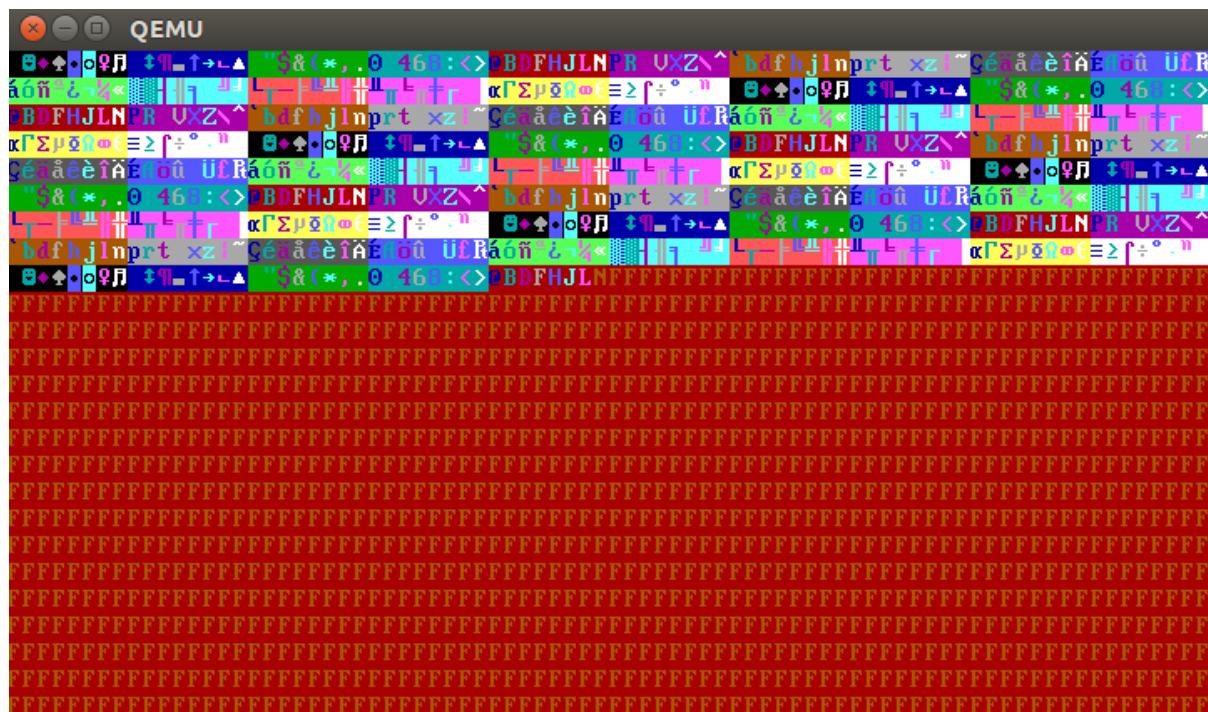
Protector de pantalla

Para configurar el protector de pantalla, se utiliza el comando `set_screensaver` [value]. Al presionar enter, el sistema mostrará un mensaje indicando que despues de [value] segundos de inactividad este se disparará:



```
QEMU
Introducir nombre de usuario (Max 20 caracteres)>Wally
Wally>set_screensaver 5
El protector de pantalla se activara despues de 5 segundos de inactividad.
Wally>_
```

Transcurrido el tiempo, se mostrará el protector de pantalla:



El estado de la pantalla volverá a la normalidad al presionar cualquier tecla.