Consolas gráficas

La API de consolas gráficas permite enviar y recibir texto por el dispositivo de vídeo por defecto. Si bien esta es la API que Atomik usa por defecto para escribir mensajes de estado por pantalla y leer datos del teclado, en una versión futura se deben redirigir todas las notificaciones del kernel a una API de logging independiente del backend (por ejemplo, enviar los mensajes de estado a un puerto serie). Nótese además que la API de consolas gráficas hace uso de secuencias de escape ANSI y ECMA, por lo tanto hay una fuerte dependencia del dispositivo de vídeo subyacente.

La API de consolas gráficas se declara en el fichero de cabecera `console/console.h`, el cual le da a la macro `SYSCON_NUM` el valor del número de consolas gráficas soportadas al mismo tiempo por Atomik.

Funciones

`void console_setup (struct console *con)`

Inicializa la estructura describiendo la consola `con` con los valores por defecto. Si la consola del sistema (es decir, la consola activa) no ha sido inicializada, esta será asignada a `con`.

Reentrante : no
Thread-safe : no

`void console_set_buffer (struct console *con, void *base)`

Establece el buffer donde se almacenará el texto de la consola. Cada caracter se define como un `schar`, una estructura de dos bytes conteniendo el código del caracter y sus atributos (color de frende y fondo). El buffer debe tener, por lo tanto, un tamaño de al menos `con->height * con->width * sizeof (schar)` bytes.

Nótese que la estructura de este buffer se hizo intencionalmente compatible con el buffer de vídeo EGA, de forma que pasando el parámetro base como `(void *) 0xb8000` en casi cualquier PC x86 nos serviría para empezar a escribir texto por pantalla.

Reentrante : sí
Thread-safe : no

`void *console_get_buffer (struct console *con)`

Devuelve la dirección del buffer de texto de una consola.

Reentrante : sí
Thread-safe : no

`int` console_set_param (`struct` console *con, `int` param, `DWORD` value)

Establece el parámetro `param` de la consola `con` a un valor `value`. Estos parámetros sirven para establecer cómo se debe comportar la consola a la hora de mostrar texto, limpiar la pantalla, enviar un caracter, etc.
La función devuelve 0 en caso de que el parámetro haya podido ser modificado, o `KERNEL_ERROR_VALUE` en caso de error. Los parámetros booleanos asumen el valor 0 como falso y distinto de 0 como verdadero. La lista de parámetros aceptados por el momento son :

CONSOLE_PARAM_EXPLICIT_CRLF

Especifica si se debe explicitar un retorno de carro (CR, carry return) y nueva línea (LF, line feed) para saltar al principio de la siguiente línea, haciendo que LF sólo produzca un incremento de la coordenada vertical actual del cursor. Este es el comportamiento por defecto de DOS. Los sistemas Unix asumen simplemente LF como el caracter para producir el salto al principio de la línea. Por defecto su valor es 0 (desactivado).

CONSOLE_PARAM_CHANGE_CUR

Indica si cada vez que se envía texto a la consola se le debe pedir al dispositivo de vídeo que mueva el cursor de la pantalla a la posición donde se escribirán los siguientes caracteres. Este parámetro es necesario sobre todo para evitar la interferencia de consolas ocultas (no asociadas a un buffer de vídeo hardware) con la consola del sistema. Su valor por defecto es 0 salvo para la consola del sistema, en la que vale 1.

CONSOLE_PARAM_CLEAR_CHAR

Indica el caracter con el que debe limpiarse una porción o la totalidad del contenido la consola ante una orden de borrado. Su valor por defecto es `' '` (espacio).

CONSOLE_PARAM_CLEAR_COLOR

Byte de atributo de color que se le debe añadir a cada caracter enviado a la consola. Por defecto es `VIDEO_ATTR (VIDEO_COLOR_WHITE, VIDEO_COLOR_BLACK)` (fondo negro, letras grises).

CONSOLE_PARAM_CR_CHAR

Especifica el caracter que debe interpretarse como retorno de carro. Su valor por defecto es `'\r'` (retorno de carro según la tabla ASCII).

CONSOLE_PARAM_LF_CHAR

Especifica el caracter que debe interpretarse como línea nueva. Su valor por defecto es `'\n'` (retorno de carro según la tabla ASCII).

CONSOLE_PARAM_BS_CHAR

Define el caracter de borrado para la consola. El envío de este caracter produce el desplazamiento del cursor a la columna anterior de la línea activa y el borrado del caracter en la misma. Su valor por defecto es `'\b'`

CONSOLE_PARAM_LOCALECHO

Define si los caracteres recibidos por el teclado deben enviarse a la consola activa. No implementado de momento. Su valor por defecto es 1 (activado).

DWORD console_get_param (struct console *con, int param)

Devuelve el valor del parámetro `param` de la consola `con`, ó 0 si el parámetro no existe. Ver `console_get_param` para más detalles.

Reentrante : sí
Thread-safe : no

void console_gotoxy (struct console *con, WORD x, WORD y)

Envía el cursor de la consola `con` a la columna `x` de la fila `y`, localizándose la esquina superior izquierda de la pantalla en `x = 0, y = 0`. Si `CONSOLE_PARAM_CHANGE_CUR` está activado, el cursor de la pantalla será desplazado igualmente a las coordenadas `x, y`.

void console_putchar (struct console *con, char c)

Envía el caracter `c` a la consola. La función `console_putchar` interpreta caracteres de control, así como secuencias de escape ECMA y ANSI, por lo que por lo que no se puede afirmar que el envío de un caracter mediante `console_putchar` siempre vaya a reflejar como un caracter imprimible en la consola.

Si el caracter no es de control (no es ni CR, ni LF, ni BS), entonces es enviado a la consola en la posición descrita por el cursor, con los atributos especificados por el parámetro `CONSOLE_PARAM_CLEAR_COLOR`.

En el caso de que el caracter `c` no sea un caracter de control y/o encuentre dentro de una secuencia de escape, el cursor incrementará la coordenada horizontal en una unidad. Si el cursor esté en el borde derecho de la consola, este se moverá al principio de la línea siguiente. Y si a su vez, el cursor esté en la última línea, la función `console_putchar` provocará el desplazamiento (*scroll*) del texto, moviendo todas las líneas a la anterior (de forma que la línea superior se perderá) y dejando la última línea en blanco. Este es el mismo comportamiento de la consola ante un caracter de nueva línea. Si el caracter es el definido por el parámetro `CONSOLE_PARAM_CR_CHAR`, el cursor se mueve a la columna 0 de la línea actual. Si el caracterer es el del parámetro `CONSOLE_PARAM_BS_CHAR` y ya se encuentra en la columna 0, será ignorado.

Si la consola con es la consola del sistema, entonces se enviará la orden de refresco al dispositivo de vídeo.

Reentrante : no
Thread-safe : no

`void console_write (struct console *con, const char *buffer, int size)`

Envía `size` caracteres contenidos en el buffer `buffer` a la consola con, interpretando secuencias de escape y caracteres de control. Su comportamiento es equivalente a ejecutar `console_putchar` por cada uno de los `size` caracterers de `buffer`. Si la consola es la consola del sistema, entonces se enviará la orden de refresco al dispositivo de vídeo.

Reentrante : no
Thread-safe : no

`void console_puts (struct console *con, const char *s)`

Envía una cadena de caracteres `s` a la consola con en formato ASCII-Z (usando el caracter nulo como marca de final de cadena), interpretando secuencias de escape y caracteres de control. La función es equivalente a llamar a `console_putchar` por cada uno de los caracteres de `s`. Si la consola es la consola del sistema, entonces se enviará la orden de refresco al dispositivo de vídeo.

Reentrante : no
Thread-safe : no

`void console_putchar_raw (struct console *con, char c)`

Envía un caracter `c` a la consola `con`, ignorando la semántica del caracter `c`. Enviar un salto de línea o un caracter de escape no producirá más cambios en el cursor que aumentar la posición horizontal, saltar de línea por desbordamiento horizontal o producir un desplazamiento por desbordamiento vertical. En su lugar, el caracter se enviará tal cual, mostrándose (en caso de que sea la consola del sistema) el caracter de la página de códigos asociado al mismo. Si la consola es la consola del sistema, entonces se enviará la orden de refresco al dispositivo de vídeo.

Reentrante : no
Thread-safe : no

`void console_write_raw (struct console *con, const char *buffer, int size)`

Envía `size` caracteres contenidos en el buffer `buffer` a la consola `con`, ignorando la semántica de los caracteres a enviar. Su comportamiento es equivalente a ejecutar `console_putchar_raw` por cada uno de los `size` caracterers de `buffer`. Si la consola es la consola del sistema, entonces se enviará la orden de refresco al dispositivo de vídeo.

Reentrante : no
Thread-safe : no

`void console_puts (struct console *con, const char *s)`
Envía una cadena de caracterers s a la consola con en formato ASCII-Z (usando el caracter nulo como marca de final de cadena), ignorando la semántica de los caracteres a enviar. La función es equivalente a llamar a `console_putchar_raw` por cada uno de los caracteres de `s`. Si la consola es la consola del sistema, entonces se enviará la orden de refresco al dispositivo de vídeo.

Reentrante : no
Thread-safe : no

`void console_clear (struct console *con)`
Limpia el contenido de la consola, rellenándol con el caracter de borrado definido mediante el parámetro `CONSOLE_PARAM_CLEAR_CHAR` y colocando el cursor en la posición `(0, 0)`. Si la consola es la consola del sistema, entonces se enviará la orden de refresco al dispositivo de vídeo.

`int console_get_current ()`
Devuelve el identificador de la consola del sistema (activa). Este será siempre un valor entre 0 y `SYSCON_NUM - 1`, representando un índice dentro del array de consolas `struct console syscon_list[]`, el cual debe ser el elemento apuntado por `struct console *syscon`. Si ninguna consola ha sido inicializada, entonces se devuelve `KERNEL_ERROR_VALUE`.

Reentrante : no
Thread-safe : no

`int console_switch (int con)`
Hace que la consola de índice con sea la consola del sistema. Cambia el parámetro `CONSOLE_PARAM_CHANGE_CUR` de la consola anterior a 0 y pone a 1 el de la nueva. Se intercambia el contenido de los búfferes de ambas consolas y se establece la dirección del nuevo al comienzo de la memoria de vídeo. El puntero `struct console *syscon` es actualizado a la dirección de la nueva consola del array `struct console syscon_list[]`. La función devuelve 0 en caso de éxito y `KERNEL_ERROR_VALUE` en caso de error.

Reentrante : no
Thread-safe : no

Variables
`extern struct console syscon_list[]`
Array de `SYSCON_NUM` elementos con la descripción de las consolas del sistema.

`extern struct console *syscon`
Dirección de la consola del sistema.