

Informe TPE J2OS

Modo protegido con GRUB

José Noriega leg. 51231
Javier Perez Cuñarro leg. 49729

Tabla de contenidos

[Introducción](#)

[Consideraciones](#)

[Shell](#)

[Terminales virtuales](#)

[Chat](#)

[Teclado](#)

[Referencias](#)

Introducción

Este sistema operativo pone en práctica los conceptos del modo protegido de Intel y el manejo de interrupciones, tanto de hardware como de software. Con estos objetivos en mente, se plantea un sistema monotarea, es decir, que corre un único proceso a la vez. La interfaz visual consiste en un shell al estilo command-line de Unix que permite escribir texto y ejecutar comandos.

Consideraciones

Shell

Se pretende emular el comportamiento de los shells que normalmente utilizamos en sistemas Unix. Al ingresar texto y presionar enter, se parsea el comando ingresado junto a sus parámetros contiguos. El parseo corre a cargo de la función `sscanf` que, a diferencia de la versión de la librería estándar de C, está hecha para levantar la primer palabra como nombre de comando (de hasta 10 letras) y aquello que sigue al primer espacio, como parámetros correspondientes al comando.

Se presenta un mensaje del tipo `"guest@J2OS-terminal-1$"` para indicar la terminal virtual que está activa (no se cuenta con manejo de usuarios, por ello siempre figura el nombre de `guest`).

Terminales virtuales

Existen 3 terminales virtuales independientes, a las cuales se accede mediante `ALT + número` de terminal. Cada una cuenta con su propio buffer de contenido y un buffer con la línea ingresada en el shell (pero no ejecutada). Esto se pensó para poder cambiar de terminal y no perder el historial actual de la pantalla, ni el texto ingresado parcialmente a ejecutar.

Chat

El modo chat se ubica como una especie de terminal "especial" en el `ALT + 4`. Al ingresar, desaparece el prompt estándar y sólo se permite ingresar texto mediante una línea al fin de pantalla, y recibir texto de otra computadora en el resto de la pantalla. La cantidad de caracteres a enviar está limitada por el tamaño de la última línea. El historial visible

corresponde a la cantidad de líneas que entra en pantalla a la vez, perdiendo mensajes anteriores. Este modo sólo funciona cuando ambas computadoras están dentro del modo chat. Si se envía un mensaje a una computadora fuera del modo chat, este mensaje se ve en la pantalla local pero nunca es recibido por la computadora remota y, por ende, nunca llega a verse en la pantalla remota.

En cuanto a la implementación elegida, para evitar congestionar el envío de caracteres mediante el puerto serie, se decidió ir enviando caracter por caracter a medida que el usuario escribe, en lugar de enviar todo de golpe al presionar enter. Esto implica que el sistema de uno de los dos participantes puede recibir "basura" (caracteres escritos que nunca serán ejecutados) y nunca mostrarlos en pantalla. Recién al detectar el arribo de un `\n` mediante el puerto serie se imprime todo lo que fue llegando con anterioridad.

Teclado

La conversión de scancodes del teclado a los códigos ascii correspondientes se logra mediante una matriz única declarada como `scancode_table[LANGUAGES][STATES][KEYS]`. Al principio parecía más claro usar varias matrices por separado, pero contar con una matriz unificada con 3 entradas resultó muy práctico. Las vocales tildadas en el caso del español, sin embargo, se procesan aparte.

Se optó por hacer el cambio de idioma del teclado con la tecla TAB para que resulte un atajo veloz para el usuario. Esto implica que el funcionamiento tradicional de la tecla TAB no fue implementado. A su vez, se decidió no utilizar las teclas del teclado numérico, las flechas, las function keys y otras cuantas teclas de función (tales como ESC, Pg. Up o Supr). El funcionamiento de las mismas no aportaba al fin de este sistema.

Referencias

- OSDev Wiki
http://wiki.osdev.org/Main_Page
- Scancodes del teclado
<http://www.win.tue.nl/~aeb/linux/kbd/scancodes.html>