

# AlamOS

## Informe

### Integrantes:

Farré Lucas Andrés 53180  
Meola Franco Román 52163

### Docente:

Vallés, Santiago Raúl  
Merovich, Horacio Víctor  
Macintosh

## Decisiones de Diseño

### Funciones de la biblioteca estándar de C

#### Tratando de seguir los estándares

En todo momento utilizamos el libro “El Lenguaje de Programación C” como nuestra guía para respetar los prototipos de todas las funciones que necesitamos escribir. En algunos casos, por la ausencia del manejo de memoria dinámica en nuestro kernel, necesitamos agregar un parámetro a algunas de esas funciones. También por la misma razón recurrimos al uso de variables globales.

#### Funciones con argumentos variables

Al momento de programar la función *printf* nos dimos cuenta que necesitábamos saber escribir funciones que reciban una cantidad de argumentos variables. Para eso, siguiendo con lo que dice K&R, conseguimos de internet el header de C *stdarg.h*. El mismo cuenta con un conjunto de macros que proporcionan recursos para recorrer una lista de argumentos de función de tamaño y tipo desconocido. Las mismas se explican en el apéndice B7 del K&R.

Estas macros nos sirvieron para hacer las siguientes funciones de la biblioteca estándar: *printf*, *fprintf*, *vfprint*, *scanf*, *fscanf*, *vfscanf*.

## Uso de Estructuras

#### Estructura *program*

Para el manejo de comandos disponibles en el shell, decidimos utilizar un arreglo de estructuras *program* que cuenta con el nombre del comando con el que se lo invocará desde el shell, su correspondiente puntero a función y una cadena de texto con una breve descripción sobre la utilidad del comando (que será utilizado por la aplicación *man*).

# Reconocimiento de Hardware

## El registro MSR

Luego de leer el manual de Intel, utilizamos la instrucción de assembler *rdmsr*, que permite leer el registro MSR del microprocesador Intel. Obtuvimos del MSR\_TEMPERATURE\_TARGET el valor de la TJ Max (que es la temperatura máxima que el procesador puede soportar) y el Digital Readout del IA32\_THERM\_STATUS para luego hacer la diferencia y obtener la temperatura real en grados centígrados.

## La base de datos PCI

En principio conseguimos de pcidatabase.com el arreglo de estructuras de vendors y devices para ser usados por nuestra función *lspci*. Nos encontramos con los siguientes problemas: en principio el .c provisto por el sitio no compilaba, por varios errores en el código. Luego de parsear la totalidad del archivo nos encontramos con que nuestro kernel no booteaba, debido a que el arreglo de estructuras era demasiado grande (más de 7000 estructuras). En la clase teórica nos dijeron que cambiando el espacio por defecto que el linker ld asigna al data segment, podríamos solucionar el problema. Igualmente, nos encontramos con que, después de todo, la base de datos había sido actualizada en el año 2010, y no detectaba correctamente nuestros dispositivos. Por ello decidimos crear un arreglo reducido con los componentes que el comando de linux *lspci* devolvió en nuestras máquinas (además de las que VirtualBox utiliza).

# Bibliografía Consultada

- **K&R**
  - Prototipos de funciones de la biblioteca estándar.
- **PCI Database**
  - [http://www.pcidatabase.com/pci\\_c\\_header.php](http://www.pcidatabase.com/pci_c_header.php)
  - Uso de la base de datos para la función *lspci*.
  - Estructura de Vendors y Devices
- **OS Dev Wiki**
  - <http://wiki.osdev.org>
  - [http://wiki.osdev.org/Text\\_UI](http://wiki.osdev.org/Text_UI) Atributos de pantalla (colores y fondos).
  - <http://wiki.osdev.org/RTC> Direcciones de puertos y datos para obtener la hora.
  - [http://wiki.osdev.org/GCC\\_Stack\\_Smashing\\_Protector](http://wiki.osdev.org/GCC_Stack_Smashing_Protector) Para saltarse la protección que el GCC por default hace sobre el uso del stack.
  - <http://wiki.osdev.org/PCI> Direcciones de puertos y registros para la función *lspci*.
- **Computer Engineering**
  - <http://www.computer-engineering.org/ps2keyboard/scancodes1.html> Tabla de makecodes y breakcodes.
- **Otros**
  - <http://bos.asmhackers.net/docs/timer/docs/cmos.pdf> Para la función *time*