

# Organización del Computador II

## Trabajo Práctico 2

Emilio Almansi  
ealmansi@gmail.com

Álvaro Machicado  
rednaxela.007@hotmail.com

Miguel Duarte  
miguelfeliped@gmail.com

2<sup>do</sup> cuatrimestre de 2013

### Resumen

Un breve abstract sobre como arruinar tu vida en 24 hs, cortesía de Bochs.

## Índice

1. Introducción	2
2. Modo real y modo protegido	3
3. Global Descriptor Table (GDT)	4
4. Paginación	5
5. Interrupciones	6
6. Manejo de tareas	7
7. Scheduler	8
8. Conclusiones	9

## 1. Introducción

## 2. Modo real y modo protegido

En la GDT hay que poner los descriptores de segmento y los descriptores de TSS para cada tarea y para cada bandera.

La misma está representada como un arreglo “*gdt\_entry*” declarado de manera global en C. Las *gdt\_entry* son structs de 4 bytes que poseen un campo para cada atributo de una entrada de gdt.

Los descriptores de segmento fueron cargados de manera estática en tiempo de compilación. Lo mismo con el descriptor de la IDT. Esto fue posible porque se conocen de antemano todos los valores necesarios para completar los descriptores.

A la hora de cargar los descriptores de TSS nos encontramos con la siguiente dificultad: Los descriptores de TSS fueron declarados como una variable global en código C. Por lo tanto en tiempo de compilación no se sabe en qué dirección van a ser cargados. Por este motivo se cargan de manera dinámica mediante una función que se llama desde *kernel.asm*. La función sencillamente crea una entrada más en el arreglo que representa de *gdt\_entry* con los atributos adecuados.

### 3. Global Descriptor Table (GDT)

En la GDT hay que poner los descriptores de segmento y los descriptores de TSS para cada tarea y para cada bandera.

La misma está representada como un arreglo “*gdt\_entry*” declarado de manera global en C. Las *gdt\_entry* son structs de 4 bytes que poseen un campo para cada atributo de una entrada de gdt.

Los descriptores de segmento fueron cargados de manera estática en tiempo de compilación. Lo mismo con el descriptor de la IDT. Esto fue posible porque se conocen de antemano todos los valores necesarios para completar los descriptores.

A la hora de cargar los descriptores de TSS nos encontramos con la siguiente dificultad: Los descriptores de TSS fueron declarados como una variable global en código C. Por lo tanto en tiempo de compilación no se sabe en que dirección van a ser cargados. Por este motivo se cargan de manera dinámica mediante una función que se llama desde kernel.asm. La función sencillamente crea una entrada más en el arreglo que representa de *gdt\_entry* con los atributos adecuados.

## 4. Paginación

## 5. Interrupciones

## 6. Manejo de tareas

## 7. Scheduler



## 8. Conclusiones