

loco-mouse

Arquitecturas de Computadoras

Informe de desarrollo

TPE PRIMER CUATRIMESTRE 2014

Colloca, Tomás	54167
Gutierrez, Ignacio	54293
Prudhomme, Franco	54263

Índice

I. Introducción	3
II. Distinción entre el espacio del usuario y el espacio del sistema operativo.....	4
III. Espacio del sistema operativo	5
III.a. Driver del teclado y del mouse.....	5
III.b. Driver de video	5
III.c. Driver del Real Time Clock (RTC)	6
III.d. Driver del PC Speaker.....	6
III.e. Terminal.....	6
IV. Espacio del usuario	7
IV.a. Librerías básicas de C.....	7
IV.a.1. Stdlib.h.....	7
IV.a.2. Ctype.h.....	7
IV.a.3. Stdio.h.....	8
IV.a.4. String.h	8
IV.a.5. Math.h	8
IV.a.6. Time.h.....	9
IV.b. Intérprete de comandos (Shell).....	9
IV.c. Interfaz grafica LoMoG (Loco Mouse GUI).....	9

I. Introducción

Se introdujo como trabajo práctico especial la implementación de un micro kernel que implemente y administre los recursos de hardware de una computadora, y muestre características del modo protegido de Intel.

II. Distinción entre el espacio del usuario y el espacio del sistema operativo

Como fue mencionado, uno de los objetivos del proyecto fue el hecho de poder simular el modo protegido que brinda Intel, separando dos grandes espacios denominados `Kernel Space` y `User Space`. Si bien la forma adoptada por Intel para poder lograr esta separación de áreas es mediante el método de paginación de memoria, en el trabajo se optó por simular aquella división haciendo que la comunicación entre ambos espacios sea únicamente mediante `syscalls`.

III. Espacio del sistema operativo

El Kernel Space es el encargado de manejar los drivers, ya que estos se comunican con el hardware. El sistema cuenta con los siguientes drivers: video, mouse, teclado, PC Speaker y real time clock.

III.a. Driver del teclado y del mouse

Tanto el driver de mouse como el driver de teclado funcionan mediante el mismo principio, ambos desconocen de la existencia de todo componente, a excepción del hardware con el cual éstos interactúan. Cada vez que dicho hardware genera una interrupción, el driver interpreta la información que éste le otorga, distribuyéndola mediante el lanzamiento de un evento. Por cada driver hay dos archivos, uno llamado **listener** (en español escuchador), el cual contiene dos estructuras: una de ellas encapsula información relevante del evento generado, mientras que la otra estructura contiene una serie de funciones que saben reaccionar ante determinados eventos (relacionados al driver en cuestión). El otro archivo es el denominado evento, el cual almacena una colección de **listeners** y cuando se lo indica recorre esta misma colección, ejecutando por cada uno la función correspondiente al evento generado.

III.b. Driver de video

El driver de video provee funciones que facilitan la interacción con el hardware de video. Internamente maneja dos arreglos de estructuras denominados **tPanels** y **tButtons**, que describen la información propia de un espacio. Dichas estructuras facilitan la segmentación de la pantalla en distintos espacios de trabajo.

III.c. Driver del Real Time Clock (RTC)

El driver del Real Time Clock ofrece una serie de funciones para solicitar la hora del sistema tanto como para modificarla.

III.d. Driver del PC Speaker

El driver del PC Speaker ofrece una serie de funciones para realizar un sonido con la frecuencia y duración deseada.

III.e. Terminal

Es una entidad que funciona como nexo entre el driver de teclado y el User Space. Ésta maneja un listener de teclado para hacer posible el arribo de información.

IV. Espacio del usuario

IV.a. Librerías básicas de C

Para el proyecto fueron implementadas ciertas librerías para simular la librería estándar de C. Éstas son: string.h, stdio.h, stdlib.h y ctype.h.

IV.a.1. Stdlib.h

```
char* itoa(int num, int base);
char* utoa(unsigned int num, int base);
char* ptoa(void* ptr);
int  atoi(char* s);
int  is_num(char* s);
int  rand(void);
void srand(unsigned new_seed);
```

IV.a.2. Ctype.h

Esta librería posee únicamente macros, es decir, que toda función presente en la librería estándar de C posee su macro equivalente.

```
int islower(int x);
int isupper(int x);
int isalpha(int x);
int isdigit(int x);
int isalnum(int x);
int isgraph(int x);
int isprint(int x);
int iscntrl(int x);
int ispunct(int x);
int isspace(int x);
int isxdigit(int x);
int isascii(int x);
```

IV.a.3. Stdio.h

```
int    getc(FILE* stream);
int    getchar();
char*  gets(char* s);
char*  fgets(char* s, int num, FILE* stream);
int    printf(const char* fmt, . . .);
int    vprintf(const char* fmt, va_list args);
int    fprintf(FILE* stream, const char* fmt, . . .);
int    vfprintf(FILE* stream, const char* fmt, va_list args);
void   putc(int c, FILE* stream);
void   putchar(int c);
void   puts(const char* s);
int    scanf(const char* fmt, . . .);
int    vscanf(const char* fmt, va_list args);
int    sscanf(const char* src, const char* fmt, ...);
int    vsscanf(const char* src, const char* fmt, va_list args);
int    read(int fd, char* buff, size_t count);
int    write(int fd, char* buff, size_t count);
```

IV.a.4. String.h

```
int    strcmp(const char* s1, const char* s2);
int    strlen(const char* s);
char*  strcpy(char* dest, const char* src);
char*  strcat(char* s1, const char* s2);
char*  strchr(char* s, int c);
```

IV.a.5. Math.h

```
int    pow(int n1, int n2);
int    round(double n);
int    floor(double n);
int    ceil(double n);
```


IV.a.6. Time.h

```
unsigned int    time();
int             time_get_seconds();
int             time_get_minutes();
int             time_get_hours();
char*          time_get_day();
int             time_get_day_num();
int             time_get_month();
int             time_get_year();
int             time_get_century();
void            time_set_time(int secs, int mins, int hour);
void            time_set_date(int day, int day_num, int
                             month, int year, int century);
void            time_set_alarm(int secs, int mins, int hour);
unsigned int    time_current_millis();
int             time_rtc(unsigned int regis_id, int* data, int op);
```

IV.b. Intérprete de comandos (Shell)

Facilita al usuario la invocación mediante comandos de los distintos programas presentes en el sistema.

IV.c. Interfaz grafica LoMoG (Loco Mouse GUI)

Ofrece la posibilidad de iniciar una simulación del modo visual del sistema junto con una serie de herramientas para la creación de paneles y botones dentro de él.