



UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERIA INFORMÁTICA

**Desarrollo de un gestor de tareas y un kernel para la arquitectura
IA-32 que sirva de apoyo durante el proceso de docencia de los
Sistemas Operativos.**

(Apéndices)

TRABAJO ESPECIAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR

Ocanto España, Victor José

Rodríguez Boscán, Aneriz del Carmen

PROFESOR GUIA

Rodolfo Campos

FECHA

Caracas, 29 de Octubre de 2008

ÍNDICE APÉNDICES

Apéndice A – Documentos de Visión y Diagrama de Clases.....	1
Documento de Visión – Módulo Arranque.....	1
Introducción.....	4
Descripción del Módulo.....	4
Definiciones, Acrónimos y Abreviaciones.....	4
Referencias.....	5
Requerimientos Funcionales.....	6
RF.1.1.1 Cargar el sistema operativo.....	6
RC 1.1.1 Cumplimiento de especificaciones.....	7
RC 1.1.2 Posibilidad de Incorporar cambios.....	7
Actores.....	8
Diagrama de Casos de Uso.....	8
Documento de Visión – Módulo Video.....	9
Introducción.....	12
Descripción del Módulo.....	12
Definiciones, Acrónimos y Abreviaciones.....	12
Referencias.....	12
Requerimientos Funcionales.....	13

Actores	RF.1.2.1 Imprimir por pantalla un carácter.....	13
Diagrama de Casos de Uso	RF.1.2.2 Imprimir por pantalla una cadena de caracteres...	13
Documento de Requisitos	RF.1.2.3 Imprimir por pantalla números enteros.....	14
Introducción	RC 1.2.1 Cumplimiento de especificaciones.....	14
	RC 1.2.2 Posibilidad de Incorporar cambios.....	14
	RC 1.2.3 Legibilidad del código.....	15
Actores.....		15
Diagrama de Casos de Uso.....		16
Documento de Visión – Módulo Interrupciones.....		17
Introducción.....		20
	Descripción del Módulo.....	20
	Definiciones, Acrónimos y Abreviaciones.....	20
Referencias.....		20
Requerimientos Funcionales.....		21
	RF.1.3.1 Capturar excepciones.....	21
	RF.1.3.2 Capturar interrupción de dispositivo.....	21
	RC 1.3.1 Cumplimiento de especificaciones.....	22
	RC 1.3.2 Posibilidad de Incorporar cambios.....	22
	RC 1.3.3 Legibilidad del código.....	22

Actores.....	23
Diagrama de Casos de Uso.....	23
Documento de Visión – Módulo Teclado.....	24
Introducción.....	27
Descripción del Módulo.....	27
Definiciones, Acrónimos y Abreviaciones.....	27
Referencias.....	27
Requerimientos Funcionales.....	28
RF.1.4.1 Leer un caracter del teclado.....	28
RF.1.4.2 Imprimir Teclas alfanuméricas.....	28
RC 1.4.1 Cumplimiento de especificaciones.....	29
RC 1.4.2 Posibilidad de Incorporar cambios.....	29
Actores.....	30
Diagrama de Casos de Uso.....	30
Documento de Visión – Módulo Reloj.....	31
Introducción.....	34
Descripción del Módulo.....	34
Definiciones, Acrónimos y Abreviaciones.....	34
Referencias.....	34

Requerimientos Funcionales.....	35
RF.2.1.1 Establecer la frecuencia de reloj.....	35
RF.2.1.2 Reconocer interrupción del reloj.....	35
RC 2.1.1 Cumplimiento de especificaciones.....	36
RC 2.1.2 Posibilidad de Incorporar cambios.....	36
Actores.....	36
Diagrama de Casos de Uso.....	37
Documento de Visión – Módulo Procesos.....	38
Introducción.....	41
Descripción del Módulo.....	41
Definiciones, Acrónimos y Abreviaciones.....	41
Referencias.....	41
Requerimientos Funcionales.....	42
RF.2.2.1 Crear un proceso.....	42
RC 2.2.1 Cumplimiento de especificaciones.....	43
RC 2.2.2 Posibilidad de Incorporar cambios.....	43
Actores.....	44
Diagrama de Casos de Uso.....	44
Documento de Visión – Módulo Planificador.....	45

Introducción.....	48
Descripción del Módulo.....	48
Definiciones, Acrónimos y Abreviaciones.....	48
Referencias.....	48
Requerimientos Funcionales.....	49
RF.2.3.1 Seleccionar el siguiente proceso a ejecutar.....	49
RC 2.3.1 Cumplimiento de especificaciones.....	50
RC 2.3.2 Posibilidad de Incorporar cambios.....	50
Actores.....	51
Diagrama de Casos de Uso.....	51
Documento de Visión – Módulo Consola.....	52
Introducción.....	55
Descripción del Módulo.....	55
Definiciones, Acrónimos y Abreviaciones.....	55
Referencias.....	55
Requerimientos Funcionales.....	56
RF.2.4.1 Interpretar los comandos.....	56
RC 2.4.1 Cumplimiento de especificaciones.....	57
RC 2.4.2 Posibilidad de Incorporar cambios.....	57

Actores.....	58
Diagrama de Casos de Uso.....	58
Diagrama de Clases.....	59
Apéndice B – Casos De Prueba.....	61
Caso de Prueba Módulo Arranque.....	61
Caso de Prueba Video.....	65
Caso de prueba Interrupciones.....	67
Caso de Prueba Módulo Teclado.....	72
Caso de Prueba Módulo Reloj.....	73
Caso de Prueba Módulo Planificador.....	74
Apéndice C – Prácticas.....	77
Práctica Módulo Video.....	77
Objetivo De La Práctica.....	77
Problema.....	77
Descripción De La Funcionalidad Pedida.....	78
Características De La Función.....	79
Otros Requerimientos.....	79
Conceptos Utilizados En La Práctica.....	79

El Entorno De Desarrollo.....	79
Solución Propuesta.....	80
Práctica Módulo Planificador.....	81
Objetivo De La Práctica.....	81
Problema.....	81
Descripción De La Funcionalidad Pedida.....	82
Características De La Función.....	82
Otros Requerimientos.....	82
Conceptos Utilizados En La Práctica.....	82
El Entorno De Desarrollo.....	82
Solución Propuesta.....	83

ÍNDICE DE FIGURAS

Figura 1 - Casos de Uso. Módulo de Arranque.....	8
Figura 2 - Casos de Uso. Módulo de Video.....	16
Figura 3 - Casos de Uso. Módulo de Interrupciones.....	23
Figura 4 - Casos de Uso. Módulo de Teclado.....	30
Figura 5 - Casos de Uso. Módulo de Reloj.....	37
Figura 6 - Casos de Uso. Módulo de Procesos.....	44
Figura 7 - Casos de Uso. Módulo del Planificador.....	51
Figura 8 - Casos de Uso. Módulo de Consola.....	58
Figura 9 - Diagrama de clases del sistema.....	59
Figura 10 - Pantalla inicial mostrada por el gestor de arranque.....	61
Figura 11 - Arranque del sistema.....	62
Figura 12 - Archivo de logs de Bochs.....	63
Figura 13 - Código inicial de la función Start.....	64
Figura 14 - Ejecución de la prueba unitaria para el video.....	65
Figura 15 - Ejecución de la prueba de división entre cero.....	68
Figura 16 - Ejecución de la prueba de acceso a un segmento no presente.....	69
Figura 17 - Ejecución de la prueba intentando acceder a un segmento nulo.....	70
Figura 18 - Ejecución de la prueba intentando acceder al segmento de datos.....	71

Figura 19 - Prueba del modulo de teclado.....	72
Figura 20 - Ejecución de la prueba del módulo de reloj.....	73
Figura 21 - Ejecución de la prueba del planificador (ejecución del proceso A)...	74
Figura 22 - Ejecución de la prueba del planificador (ejecución del proceso B)....	75
Figura 23 - Ejecución de la prueba del planificador (ejecución de ambos procesos).....	76
Figura 24 - Planificador2.h.....	83
Figura 25 - Planificador2.cpp.....	84
Figura 26 – Makefile.....	85
Figura 27 - Main.cpp.....	86
Figura 28 - Main.cpp.....	87
Figura 29 – Makefile.....	88
Figura 30 - Ejecución del código (primer proceso).....	89
Figura 31 - Ejecución del código (segundo proceso).....	90

Apéndice A

Documentos de Visión
y Diagrama de Casos de Uso

Vianyx - 1er Entregable

Visión del Módulo Arranque (Mód - 1.1)

Versión 1.0

Lista de Cambios

Número orden	Fecha	Versión	Descripción	Estado	Autor
1	05/03/2008	1.0	inicio	En revisión	Víctor Ocanto.
2	10/03/2008	1.0	Corrección de casos de uso	En revisión	Víctor Ocanto.
3	12/03/2008	1.0	Cambio en las Definiciones, Acrónimos y Abreviaciones	En revisión	Aneriz Rodriguez.

Tabla de Contenido

Introducción	4
Descripción del Módulo.....	4
Definiciones, Acrónimos y Abreviaciones	4
Referencias	5
Requerimientos Funcionales.....	6
RF.1.1.1 Cargar el sistema operativo.....	6
RC 1.1.1 Cumplimiento de especificaciones.....	7
RC 1.1.2 Posibilidad de Incorporar cambios	7
Actores.....	8
Diagrama de Casos de Uso.....	8

Introducción

Este documento proporciona una vista funcional del Módulo de Arranque del sistema Vianyx, identificando sus funcionalidades principales y complementarias, aspectos de diseño, restricciones y otros aspectos basados en los diagramas y definiciones de los casos de uso.

Descripción del Módulo

El Subsistema Arranque tiene la finalidad de cargar en memoria al núcleo de Vianyx, este módulo permitirá seleccionar el núcleo a cargar, modificar parámetros, y proporcionar el enunciado de las prácticas correspondientes.

El Módulo de Arranque es una parte fundamental para el funcionamiento del sistema Vianyx (aunque no forma parte del mismo), debido a que es el encargado de colocar en memoria y dar el control al sistema Vianyx y además con la información que se obtiene a través de él se establecerán las estructuras necesarias.

Definiciones, Acrónimos y Abreviaciones

A continuación se describen los términos, acrónimos, y abreviaciones necesarias para interpretar de forma apropiada el presente documento.

- **Gestor de Arranque:** Es un programa que te permite elegir, en el momento de arrancar el ordenador, con qué Sistema Operativo quieres trabajar. Una vez eliges uno, arranca éste y el otro Sistema Operativo "desaparece" del disco. En realidad no desaparece, sino que el gestor de arranque se encarga de ocultar su partición para hacerla invisible. Así, ambos Sistemas Operativos nunca se solapan e interfieren uno con el otro. De esta forma, con sólo reiniciar y entrar en el otro ya podremos seguir trabajando.
- **Grub: (GRand Unified Bootloader)** es un gestor de arranque múltiple que se usa comúnmente para iniciar uno o más sistemas operativos instalados en un mismo ordenador y permite elegir cual sistema operativo iniciar.
- **Núcleo:** Es el software responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma más básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, el núcleo también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado. Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad y proporciona una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso para el programador.

Referencias

- http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29
- <http://www.proyectoedora.org/argentina/?p=63>
- http://www.hugorodriguez.com/cursos/arranque_dual_01.htm

Los requerimientos funcionales, representan todas aquellas funciones básicas y primordiales de un módulo. En esta sección se presenta un breve resumen que permitirá entender el requerimiento y hacerle seguimiento a lo largo del desarrollo del sistema. Se muestra la referencia de requerimiento, impacto, descripción del requerimiento, prioridad, nivel de desempeño (desempeño/seguridad), relación con otros requerimientos o módulos, observaciones y/o acciones.

* Cargar el sistema operativo	
Requerimiento	RF 1.1.1 Cargar el sistema operativo
Impacto/Descripción	Este requerimiento define la función de cargar el sistema operativo en el hardware del computador.
Justificación	A continuación se detallan los requisitos para la implementación de este requerimiento:
Requisitos de software	• Sistema operativo • BIOS del computador • Disco duro con capacidad suficiente para el sistema operativo • Ubicación de la instalación de los archivos del sistema operativo • Puerto de conexión a internet
Impacto	Funcionalidad
Requisitos de hardware	Ninguno
Requisitos de seguridad	• El módulo de sistema operativo debe ser seguro y confiable.
Requisitos de rendimiento	
Requisitos de interfaz	
Observaciones	• Puede ser necesario configurar el BIOS.

Especificación de Requerimientos

Requerimientos Funcionales

Los requerimientos funcionales representan todas aquellas funciones básicas y primordiales del módulo. En esta sección se presenta un breve resumen que permitirá entender el requerimiento y hacerle seguimiento a lo largo del desarrollo del sistema. Se describe la referencia del requerimiento, impacto, descripción, tipo (funcionalidad, confiabilidad, seguridad, desempeño, usabilidad), relación con otros requerimientos o módulos, observaciones y/o soporte:

• <i>Cargar el sistema operativo</i>	
Referencia	RF.1.1.1 Cargar el sistema operativo
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Esta funcionalidad deberá permitir cargar el núcleo del sistema operativo seleccionado. A continuación se muestran los datos básicos necesarios • Nombre del núcleo • Ubicación de memoria a ser cargado • Dispositivo donde se encuentra almacenado • Ubicación en el sistema de archivos (ruta) • Parámetros de carga
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	• El núcleo del sistema es cargado satisfactoriamente en la ubicación de memoria asignada.
Comportamiento	
Relaciones	
Observaciones y/o soporte	• Puede resultar útil comprobar el estado de los registros

Requerimientos No Funcionales

Los requerimientos no funcionales asociados al módulo en desarrollo, son listados en ésta sección.

Referencia	RC 1.1.1 Cumplimiento de especificaciones.
Requerimiento o Descripción	El Sistema debe cumplir con todas las especificaciones propuestas.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 1.1.2 Posibilidad de Incorporar cambios
Requerimiento o Descripción	El Sistema debe tener la disponibilidad y facilidad de incorporar cambios y nuevas funcionalidades al mismo.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 1.1.3 Legibilidad del código
Requerimiento o Descripción	El código del sistema debe ser legible.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Actores.

Luego de un estudio entre las partes interesadas en el desarrollo del sistema se elaboró una lista de los actores que darán uso al sistema, estos se presentan a continuación.

Actores	Descripción
GRUB	Gestor de arranque elegido para el módulo.

Luego de identificar los usuarios del sistema se continúa con la lista de los casos de uso, los cuales se obtuvieron por medio de un estudio de todos los procesos y eventos que realizan los usuarios dentro de la aplicación.

Para la presentación de los casos de uso se toma como referencia la notación que se ha estado usando hasta el momento (UML).

Diagrama de Casos de Uso

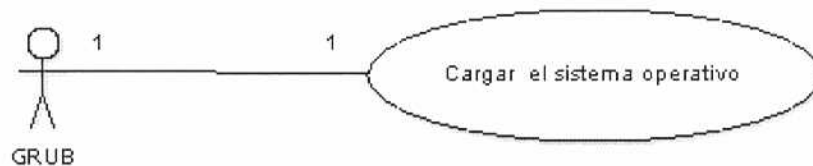


Figura 1 - Casos de Uso. Módulo de Arranque
Elaboración Propia.

Vianyx - 1er Entregable

Visión del Módulo Video (Mód - 1.2)

Versión 1.0

Número orden
1
2
3
4

Lista de Cambios

Número orden	Fecha	Versión	Descripción	Estado	Autor
1	18/03/2008	1.0	inicio	En Revisión	Aneriz Rodríguez
2	9/04/2008	1.0	Cambios en los requerimientos funcionales	En Revisión	Aneriz Rodríguez
3	12/04/2008	1.0	Cambio en diagrama de casos de uso	En Revisión	Víctor Ocanto
4	20/04/2008	1.0	Cambio en los casos de uso	En Revisión	Aneriz Rodríguez

Tabla de Contenido

Introducción	12
Descripción del Módulo.....	12
Definiciones, Acrónimos y Abreviaciones	12
Referencias	12
Requerimientos Funcionales.....	13
RF.1.2.1 Imprimir por pantalla un caracter.....	13
RF.1.2.2 Imprimir por pantalla una cadena de caracteres.....	13
RF.1.2.3 Imprimir por pantalla números enteros	14
RC 1.2.1 Cumplimiento de especificaciones.....	14
RC 1.2.2 Posibilidad de Incorporar cambios	14
RC 1.2.3 Legibilidad del código	15
Actores.....	15
Diagrama de Casos de Uso.....	16

Definiciones, Acrónimos y Abreviaciones

A continuación se definen los términos, acrónimos y abreviaciones más relevantes para interpretar el contenido de este documento.

- **Controlador:** Es el encargado de controlar el funcionamiento de los dispositivos de hardware y software.

Referencias

- **ISO 9001:** Norma internacional de gestión de la calidad.

Introducción

Este documento proporciona una vista funcional del módulo de Video del 1er entregable de Vianyx, identificando sus funcionalidades principales y complementarias, aspectos de diseño, restricciones y otros aspectos basados en los diagramas y definiciones de los casos de uso.

Descripción del Módulo

El Módulo de Video del 1er entregable de Vianyx tiene la finalidad de explicar, definir y dar los detalles más importantes acerca de la implementación del video en el sistema, explicar sus definiciones y como debería actuar el sistema.

Este módulo se encarga de controlar el hardware de video, permitirá mostrar por pantalla distintos mensajes para el usuario.

El Módulo de Video es una parte fundamental dentro del 1er entregable de Vianyx, debido a que con las funcionalidades que éste provee, puede verse por pantalla el comportamiento del sistema y se puede generar una retroalimentación de la aplicación.

Definiciones, Acrónimos y Abreviaciones

A continuación se describen los términos, acrónimos, y abreviaciones necesarias para interpretar de forma apropiada el presente documento.

- **Controlador:** pequeño programa cuya función es controlar el funcionamiento de un dispositivo del ordenador bajo un determinado sistema operativo.

Referencias

- <http://www.conozcasuhardware.com/diccio/index.htm>

Especificación de Requerimientos

Requerimientos Funcionales

Los requerimientos funcionales representan todas aquellas funciones básicas y primordiales del módulo. En esta sección se presenta un breve resumen que permitirá entender el requerimiento y hacerle seguimiento a lo largo del desarrollo del sistema se describe la referencia del requerimiento, impacto, descripción, tipo (funcionalidad, confiabilidad, seguridad, desempeño, usabilidad), relación con otros requerimientos o módulos, observaciones y/o soporte:

• <i>Imprimir por pantalla un caracter</i>	
Referencia	RF.1.2.1 Imprimir por pantalla un caracter
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Esta funcionalidad deberá permitir mostrar un carácter por pantalla. A continuación se muestran los datos básicos que deberán ser mostrados: - El carácter a ser mostrado. - Lugar en la pantalla donde será mostrado el carácter.
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	Se logra imprimir por pantalla el carácter indicado.
Comportamiento	
Relaciones	
Observaciones y/o soporte	Puede resultar útil poder indicar el color del carácter y del fondo de la pantalla.

• <i>Imprimir por pantalla una cadena de caracteres</i>	
Referencia	RF.1.2.2 Imprimir por pantalla una cadena de caracteres
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Esta funcionalidad deberá permitir mostrar una cadena de caracteres por pantalla. A continuación se muestran los datos básicos que deberán ser mostrados: - La cadena de caracteres a ser mostrada. - Lugar en la pantalla donde será mostrada la cadena de caracteres.
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	Se logra imprimir por pantalla la cadena de caracteres.
Comportamiento	
Relaciones	RF.1.2.1
Observaciones y/o soporte	- Puede resultar útil poder indicar el color de la cadena de caracteres y el color del fondo de la pantalla. - Puede resultar útil poder indicar el formato de la salida.

• Imprimir por pantalla números enteros	
Referencia	RF.1.2.3 Imprimir por pantalla números enteros
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Esta funcionalidad deberá permitir mostrar un número entero sin signo por pantalla. A continuación se muestran los datos básicos que deberán ser mostrados: • El número entero a ser mostrado. • Lugar en la pantalla donde será mostrado el número entero.
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	• Se logra imprimir por pantalla el número entero.
Comportamiento	
Relaciones	
Observaciones y/o soporte	• Puede resultar útil poder indicar el color del número entero a ser mostrado y el color del fondo de la pantalla. • Puede resultar útil poder indicar el formato de la salida. • Puede resultar útil mostrar el signo del número entero. • Puede resultar útil mezclar la salida por pantalla de números enteros y caracteres.

Requerimientos No Funcionales

Los requerimientos no funcionales asociados al módulo en desarrollo, son listados en ésta sección.

Referencia	RC 1.2.1 Cumplimiento de especificaciones.
Requerimiento o Descripción	El Sistema debe cumplir con todas las especificaciones propuestas.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 1.2.2 Posibilidad de Incorporar cambios
Requerimiento o Descripción	El Sistema debe tener la disponibilidad y facilidad de incorporar cambios y nuevas funcionalidades al mismo.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 1.2.3 Legibilidad del código
Requerimiento o Descripción	El código del sistema debe ser legible.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Actores.

Luego de un estudio entre las partes interesadas en el desarrollo del sistema se elaboró una lista de los actores que darán uso al sistema, estos se presentan a continuación.

Actores	Descripción
Sistema	Encargado de colocar los caracteres en pantalla

Luego de identificar los usuarios del sistema se continúa con la lista de los casos de uso, los cuales se obtuvieron por medio de un estudio de todos los procesos y eventos que realizan los usuarios dentro de la aplicación.

Para la presentación de los casos de uso se toma como referencia la notación que se ha estado usando hasta el momento (UML).

Diagrama de Casos de Uso

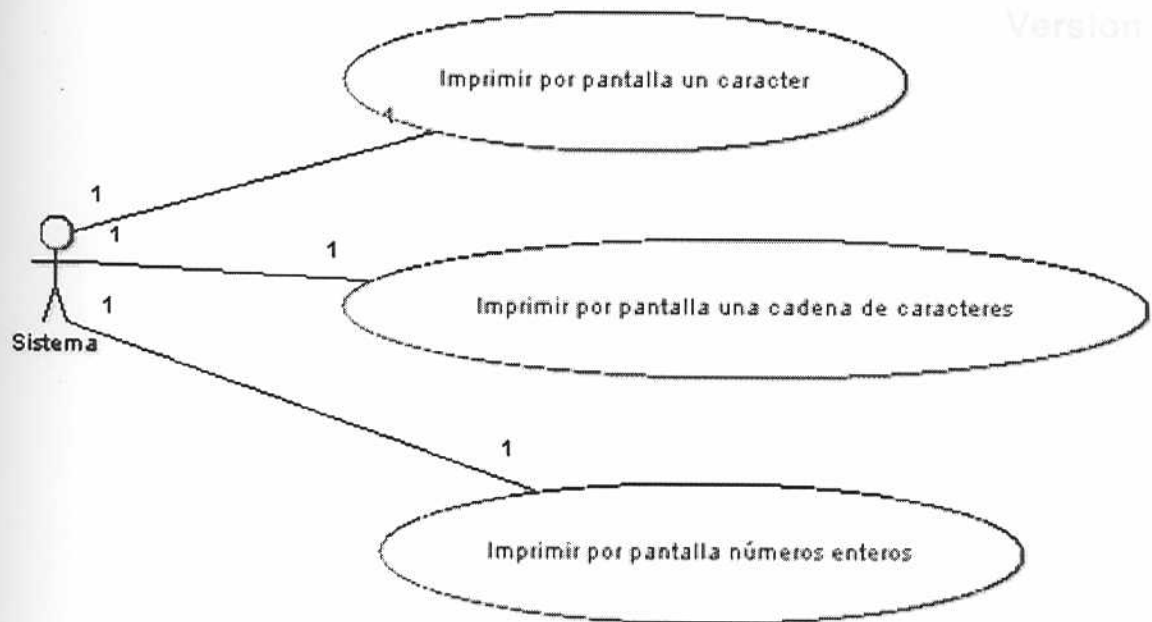


Figura 2 - Casos de Uso. Módulo de Video
Elaboración Propia.

Vianyx - 1er Entregable

Visión del Módulo Interrupciones (Mód - 1.3)

Versión 1.0

Lista de Cambios

Número orden	Fecha	Versión	Descripción	Estado	Autor
1	24/04/2008	1.0	Inicio	En Revisión	Aneriz Rodríguez
2	29/04/2008	1.0	Revisión	En Revisión	Víctor Ocanto
2	10/05/2008	1.0	Revisión	En Revisión	Víctor Ocanto
2	13/05/2008	1.0	Revisión	En Revisión	Aneriz Rodríguez

Tabla de Contenido

Introducción	20
Descripción del Módulo.....	20
Definiciones, Acrónimos y Abreviaciones	20
Referencias	20
Requerimientos Funcionales.....	21
RF.1.3.1 Capturar excepciones	21
RF.1.3.2 Capturar interrupción de dispositivo	21
RC 1.3.1 Cumplimiento de especificaciones.....	22
RC 1.3.2 Posibilidad de Incorporar cambios	22
RC 1.3.3 Legibilidad del código	22
Actores.	23
Diagrama de Casos de Uso.....	23

Introducción

Este documento proporciona una vista funcional del módulo de Interrupciones del sistema Vianyx, identificando sus funcionalidades principales y complementarias, aspectos de diseño, restricciones y otros aspectos basados en los diagramas y definiciones de los casos de uso.

Descripción del Módulo

El Módulo de Interrupciones del 1er entregable de Vianyx tiene la finalidad de explicar, definir y dar los detalles más importantes acerca de la implementación de las interrupciones en el sistema, explicar sus definiciones y como debería actuar el sistema.

Este módulo se encarga de controlar las Interrupciones del Sistema.

Definiciones, Acrónimos y Abreviaciones

A continuación se describen los términos, acrónimos, y abreviaciones necesarias para interpretar de forma apropiada el presente documento.

- **GDT:** Tabla de descriptores global. Cada segmento del área global está definido por un descriptor, existiendo una tabla llamada Tabla De Descriptores Globales o Tabla Global De Descriptores (GDT), que contiene todos los descriptores del área global. Contiene la información (descriptores) referente a los diferentes segmentos existentes en el sistema.
- **IDT:** La tabla de descriptores de interrupción (IDT) contiene los vectores de interrupción, junto con información sobre permisos, inhabilitación de interrupciones, entre otros. En los descriptores de interrupción hay selectores sobre la tabla GDT o LDT.
- **PIC8259:** Este chip admite hasta 8 señales de interrupción, pero también admite ser configurado en cascada (sus entradas proceden a su vez de otro controlador) de modo que un conjunto de controladores 8259 pueden gestionar hasta un máximo de 8*8 señales de interrupción procedentes de otros tantos dispositivos de entrada salida.

Referencias

- <http://sop.upv.es/es/es/t2-arquitectura/gen-t2-arquitectura.html>
- <http://64.233.169.104/search?q=cache:9g7JxvHmHnIJ:serdis.dis.ulpgc.es/~itis-byp/NotasDeClase/informacion/SISTEMA%2520x86/Controlador%2520Int%25208259.doc+pic+8259&hl=es&ct=clnk&cd=1&gl=ve&client=firefox-a>
- <http://www.sleepersbrains.com/tutorials/saccopharynx/Ring0/TutorialR003/R003.htm>

Especificación de Requerimientos

Requerimientos Funcionales

Los requerimientos funcionales representan todas aquellas funciones básicas y primordiales del módulo. En esta sección se presenta un breve resumen que permitirá entender el requerimiento y hacerle seguimiento a lo largo del desarrollo del sistema. Se describe la referencia del requerimiento, impacto, descripción, tipo (funcionalidad, confiabilidad, seguridad, desempeño, usabilidad), relación con otros requerimientos o módulos, observaciones y/o soporte:

• <i>Capturar excepciones</i>	
Referencia	RF.1.3.1 Capturar excepciones
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Esta funcionalidad manejar las excepciones del sistema. A continuación se muestran los datos básicos necesarios Registros de estado del procesador
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	a)
Comportamiento	Ante alguna excepción el sistema mostrara un mensaje y el numero de la excepción ocurrida y el sistema termina su ejecución
Relaciones	
Observaciones y/o soporte	•

• <i>Capturar interrupción de dispositivo</i>	
Referencia	RF.1.3.2 Capturar interrupción de dispositivo
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Esta funcionalidad manejar las interrupciones de los siguientes dispositivos - Reloj - Teclado A continuación se muestran los datos básicos necesarios Registros de estado del procesador
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	
Comportamiento	Ante la interrupción producida por alguno de los dispositivos mencionados anteriormente el sistema deberá invocar el objeto manejador correspondiente al dispositivo
Relaciones	
Observaciones y/o soporte	

Especificación de Requerimientos

Los requerimientos no funcionales asociados al módulo en desarrollo, son listados en ésta sección.

Referencia	RC 1.3.1 Cumplimiento de especificaciones.
Requerimiento o Descripción	El Sistema debe cumplir con todas las especificaciones propuestas.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 1.3.2 Posibilidad de Incorporar cambios
Requerimiento o Descripción	El Sistema debe tener la disponibilidad y facilidad de incorporar cambios y nuevas funcionalidades al mismo.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 1.3.3 Legibilidad del código
Requerimiento o Descripción	El código del sistema debe ser legible.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Actores.

Luego de un estudio entre las partes interesadas en el desarrollo del sistema se elaboró una lista de los actores que darán uso al sistema, estos se presentan a continuación.

Actores	Descripción
Módulo de Interrupciones	EL encargado de manejar las interrupciones y excepciones ocurridas

Luego de identificar los usuarios del sistema se continúa con la lista de los casos de uso, los cuales se obtuvieron por medio de un estudio de todos los procesos y eventos que realizan los usuarios dentro de la aplicación.

Para la presentación de los casos de uso se toma como referencia la notación que se ha estado usando hasta el momento (UML).

Diagrama de Casos de Uso

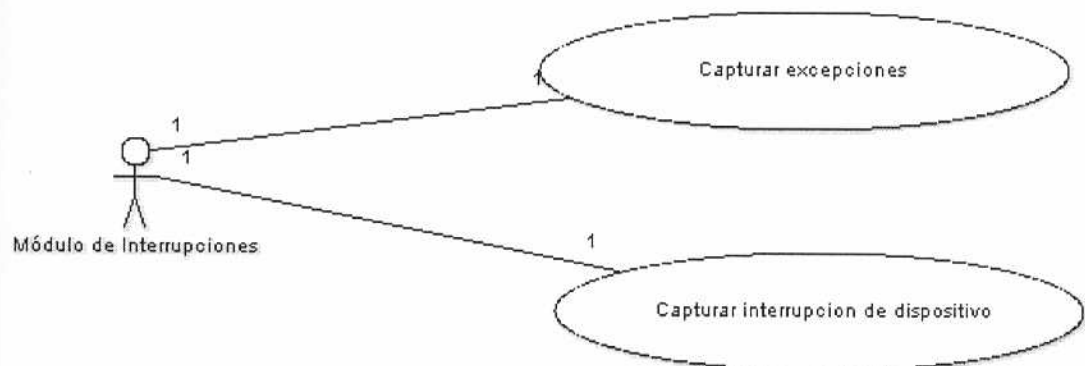


Figura 3 - Casos de Uso. Módulo de Interrupciones
Elaboración Propia.

Vianyx - 1er Entregable

Visión del Módulo Teclado (Mód - 1.4)

Versión 1.0

Número orden
1
2
3
4
5
6
7
8
9
10

Lista de Cambios

Número orden	Fecha	Versión	Descripción	Estado	Autor
1	15/05/2008	1.0	Inicio	En Revisión	Aneriz Rodríguez
1	20/05/2008	1.0	revisión	En Revisión	Aneriz Rodríguez
1	22/05/2008	1.0	revisión	En Revisión	Aneriz Rodríguez

Tabla de Contenido

Introducción	27
Descripción del Módulo.....	27
Definiciones, Acrónimos y Abreviaciones	27
Referencias	27
Requerimientos Funcionales	28
RF.1.4.1 Leer un caracter del teclado.....	28
RF.1.4.2 Imprimir Teclas alfanuméricas	28
RC 1.4.1 Cumplimiento de especificaciones.....	29
RC 1.4.2 Posibilidad de Incorporar cambios	29
Actores.	30
Diagrama de Casos de Uso.....	30

Introducción

Este documento proporciona una vista funcional del Módulo de Teclado del 1er entregable de Vianyx, identificando sus funcionalidades principales y complementarias, aspectos de diseño, restricciones y otros aspectos basados en los diagramas y definiciones de los casos de uso.

Descripción del Módulo

El Módulo de Teclado del 1er entregable de Vianyx tiene la finalidad de explicar, definir y dar los detalles más importantes acerca de la implementación del teclado en el sistema, explicar sus definiciones y como debería actuar el sistema.

Este módulo se encarga de controlar el hardware de teclado, permitirá mostrar por pantalla distintos mensajes escritos por el usuario.

El Módulo de Teclado es una parte fundamental dentro del 1er entregable de Vianyx, debido a que con las funcionalidades que éste provee, puede verse por pantalla el comportamiento del sistema y se puede generar una retroalimentación de la aplicación.

Definiciones, Acrónimos y Abreviaciones

A continuación se describen los términos, acrónimos, y abreviaciones necesarias para interpretar de forma apropiada el presente documento.

- **Controlador:** pequeño programa cuya función es controlar el funcionamiento de un dispositivo del ordenador bajo un determinado sistema operativo.

Referencias

- <http://www.conozcasuhardware.com/diccio/index.htm>

Especificación de Requerimientos

Requerimientos Funcionales

Los requerimientos funcionales representan todas aquellas funciones básicas y primordiales del módulo. En esta sección se presenta un breve resumen que permitirá entender el requerimiento y hacerle seguimiento a lo largo del desarrollo del sistema se describe la referencia del requerimiento, impacto, descripción, tipo (funcionalidad, confiabilidad, seguridad, desempeño, usabilidad), relación con otros requerimientos o módulos, observaciones y/o soporte:

• <i>Leer un carácter del teclado</i>	
Referencia	RF.1.4.1 Leer un caracter del teclado
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Esta funcionalidad deberá permitir mostrar por pantalla un carácter introducido por teclado. A continuación se muestran los datos básicos que deberán ser mostrados: • El carácter a ser leído.
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	• Se logra imprimir por pantalla el carácter indicado por teclado.
Comportamiento	Procesa la interrupción producida por la pulsación de una tecla lo transforma a ascii y lo imprime por pantalla
Relaciones	
Observaciones y/o soporte	

• <i>Imprimir Teclas alfanuméricas</i>	
Referencia	RF.1.4.2 Imprimir Teclas alfanuméricas
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Esta funcionalidad deberá permitir mostrar por pantalla teclas numéricas y de letras. A continuación se muestran los datos básicos que deberán ser mostrados: • Las teclas que sean tecleadas. • Lugar en la pantalla donde será mostrada la cadena de caracteres.
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	• Se logra imprimir por pantalla las teclas elegidas.
Comportamiento	
Relaciones	
Observaciones y/o soporte	

Requerimientos No Funcionales

Los requerimientos no funcionales asociados al módulo en desarrollo, son listados en ésta sección.

Referencia	RC 1.4.1 Cumplimiento de especificaciones.
Requerimiento o Descripción	El Sistema debe cumplir con todas las especificaciones propuestas.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 1.4.2 Posibilidad de Incorporar cambios
Requerimiento o Descripción	El Sistema debe tener la disponibilidad y facilidad de incorporar cambios y nuevas funcionalidades al mismo.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 1.4.3 Legibilidad del código
Requerimiento o Descripción	El código del sistema debe ser legible.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Actores.

Luego de un estudio entre las partes interesadas en el desarrollo del sistema se elaboró una lista de los actores que darán uso al sistema, estos se presentan a continuación.

Actores	Descripción
Sistema	Representa el encargado del procesamiento directo de la interrupción, y el cambio de estado del teclado

Luego de identificar los usuarios del sistema se continúa con la lista de los casos de uso, los cuales se obtuvieron por medio de un estudio de todos los procesos y eventos que realizan los usuarios dentro de la aplicación.

Para la presentación de los casos de uso se toma como referencia la notación que se ha estado usando hasta el momento (UML).

Diagrama de Casos de Uso

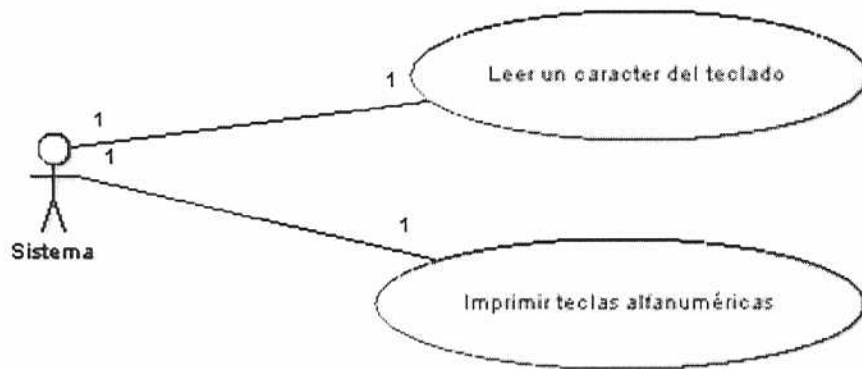


Figura 4 - Casos de Uso. Módulo de Teclado

Elaboración Propia

Vianyx – 2do Entregable

Visión del Módulo Reloj (Mód - 2.1)

Versión 1.0

Numero orden
1
2
3

Lista de Cambios

Número orden	Fecha	Versión	Descripción	Estado	Autor
1	10/07/2008	1.0	inicio	En Revisión	Aneriz Rodríguez
2	15/07/2008	1.0	Cambios de formato	En Revisión	Víctor Ocanto
3	20/07/2008	1.0	Cambios del diagrama de clases		Aneriz Rodríguez

Tabla de Contenido

Introducción	34
Descripción del Módulo.....	34
Definiciones, Acrónimos y Abreviaciones	34
Referencias	34
Requerimientos Funcionales.....	35
RF.2.1.1 Establecer la frecuencia de reloj.....	35
RF.2.1.2 Reconocer interrupción del reloj	35
RC 2.1.1 Cumplimiento de especificaciones.....	36
RC 2.1.2 Posibilidad de Incorporar cambios	36
Actores.	36
Diagrama de Casos de Uso.....	37

Introducción

Este documento proporciona una vista funcional del Módulo de Reloj del 2do entregable de Vianyx, identificando sus funcionalidades principales y complementarias, aspectos de diseño, restricciones y otros aspectos basados en los diagramas y definiciones de los casos de uso.

Descripción del Módulo

El Módulo de Reloj del 2do entregable de Vianyx tiene la finalidad de explicar, definir y dar los detalles más importantes acerca del reconocimiento del reloj en el sistema, explicar sus definiciones y como debería actuar el sistema.

Este módulo se encarga de controlar los *ticks* o ciclos del reloj, permite mostrar por pantalla cada *tick* de reloj al determinar cuando ocurre.

El Módulo de Reloj es una parte fundamental dentro del 2do entregable de Vianyx, debido a que con las funcionalidades que éste provee, se pueden generar interrupciones, intervalos de tiempo regulares y puede realizarse la planificación de procesos.

Definiciones, Acrónimos y Abreviaciones

A continuación se describen los términos, acrónimos, y abreviaciones necesarias para interpretar de forma apropiada el presente documento.

- **Controlador:** pequeño programa cuya función es controlar el funcionamiento de un dispositivo del ordenador bajo un determinado sistema operativo.
- **Ciclo de reloj:** También denominado ciclos por segundo o frecuencia, este término hace referencia a la velocidad del procesador incorporado en la CPU del ordenador, y se mide en megahertzios (MHz). A mayor índice de frecuencia, más rápido es el procesador y, en consecuencia, el ordenador. Hace unos años era frecuente encontrar procesadores a 16 MHz, pero hoy en día lo normal es que éstos superen los 120 MHz.

Referencias

- <http://www.conozcasuhardware.com/diccio/index.htm> (Julio, 2008)
- <http://www.mastermagazine.info/termino/4250.php> (Julio, 2008)

Especificación de Requerimientos

Requerimientos Funcionales

Los requerimientos funcionales representan todas aquellas funciones básicas y primordiales del módulo. En esta sección se presenta un breve resumen que permitirá entender el requerimiento y hacerle seguimiento a lo largo del desarrollo del sistema se describe la referencia del requerimiento, impacto, descripción, tipo (funcionalidad, confiabilidad, seguridad, desempeño, usabilidad), relación con otros requerimientos o módulos, observaciones y/o soporte:

• <i>Establecer la frecuencia de reloj</i>	
Referencia	RF.2.1.1 Establecer la frecuencia de reloj
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Permitirá establecer un "divisor" el cual indicará la frecuencia de activación de la interrupción de reloj.
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	
Comportamiento	
Relaciones	
Observaciones y/o soporte	

• <i>Reconocer interrupción del reloj</i>	
Referencia	RF.2.1.2 Reconocer interrupción del reloj
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Permitirá reconocer y procesar según los requerimientos específicos la interrupción de reloj, en este caso se deberá activar el planificador cuando el número de ticks sea igual al valor definido en la constante QUANTUM, y luego deberá restablecer el número de ticks a cero.
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	• Debe activarse el planificador en intervalos establecidos por la variable QUANTUM
Comportamiento	
Relaciones	
Observaciones y/o soporte	• Esta capacidad varia según el tipo de planificación elegida

Requerimientos No Funcionales

Los requerimientos no funcionales asociados al módulo en desarrollo, son listados en ésta sección.

Referencia	RC 2.1.1 Cumplimiento de especificaciones.
Requerimiento o Descripción	El Sistema debe cumplir con todas las especificaciones propuestas.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 2.1.2 Posibilidad de Incorporar cambios
Requerimiento o Descripción	El Sistema debe tener la disponibilidad y facilidad de incorporar cambios y nuevas funcionalidades al mismo.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 2.1.3 Legibilidad del código
Requerimiento o Descripción	El código del sistema debe ser legible.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Actores.

Luego de un estudio entre las partes interesadas en el desarrollo del sistema se elaboró una lista de los actores que darán uso al sistema, estos se presentan a continuación.

Actores	Descripción
Sistema	Modulo encargado del procesamiento de la interrupción de reloj

Luego de identificar los usuarios del sistema se continúa con la lista de los casos de uso, los cuales se obtuvieron por medio de un estudio de todos los procesos y eventos que

realizan los usuarios dentro de la aplicación.

Para la presentación de los casos de uso se toma como referencia la notación que se ha estado usando hasta el momento (UML).

Diagrama de Casos de Uso

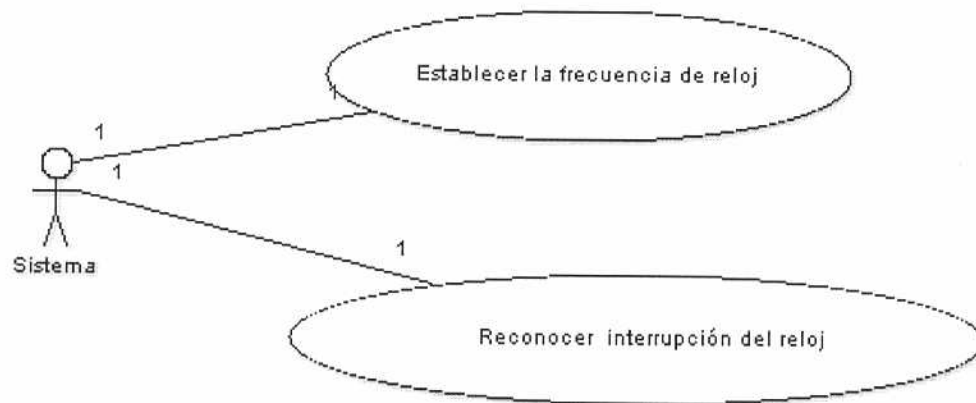


Figura 5 - Casos de Uso. Módulo de Reloj

Elaboración Propia.

Vianyx – 2do Entregable

Lista de Cambios

Visión del Módulo Procesos (Mód - 2.2)

Número orden	Fecha	Version	Descripción	En	Revisión	Por
1	22/07/2008	1.0	Inicio	En	Revisión	Araceli Rodriguez
2	24/07/2008	1.1	Cambios de formato	En	Revisión	Victor Ocampo
3	24/07/2008	1.2	Cambios del diagrama de flujo	En	Revisión	Araceli Rodriguez

Versión 1.0

Lista de Cambios

Número orden	Fecha	Versión	Descripción	Estado	Autor
1	22/07/2008	1.0	inicio	En Revisión	Aneriz Rodríguez
2	24/07/2008	1.0	Cambios de formato	En Revisión	Víctor Ocanto
3	24/07/2008	1.0	Cambios del diagrama de clases		Aneriz Rodríguez

Tabla de Contenido

Introducción	41
Descripción del Módulo.....	41
Definiciones, Acrónimos y Abreviaciones	41
Referencias	41
Requerimientos Funcionales	42
RF.2.2.1 Crear un proceso	42
RC 2.2.1 Cumplimiento de especificaciones	43
RC 2.2.2 Posibilidad de Incorporar cambios	43
Actores	44
Diagrama de Casos de Uso.....	44

Especificación de requerimientos de Software

Introducción

Este documento proporciona una vista funcional del Módulo de Procesos del 2do entregable de Vianyx, identificando sus funcionalidades principales y complementarias, aspectos de diseño, restricciones y otros aspectos basados en los diagramas y definiciones de los casos de uso.

Descripción del Módulo

El Módulo de Procesos del 2do entregable de Vianyx tiene la finalidad de explicar, definir y dar los detalles más importantes acerca de la implementación de los procesos en el sistema, explicar sus definiciones y como debería actuar el sistema.

Este módulo se encarga de la creación de un proceso lo que en este caso se refiere al establecimiento de los valores del BCP (Bloque de control de procesos) y la inclusión de dicho bloque en el arreglo del sistema.

El Módulo de Procesos es una parte fundamental dentro del 2do entregable de Vianyx, debido a que con las funcionalidades que éste provee, se puede realizar el módulo del planificador.

Definiciones, Acrónimos y Abreviaciones

A continuación se describen los términos, acrónimos, y abreviaciones necesarias para interpretar de forma apropiada el presente documento.

- **Proceso:** es un concepto manejado por el sistema operativo que consiste en el conjunto formado por:
 - Las instrucciones de un programa destinadas a ser ejecutadas por el microprocesador.
 - Su estado de ejecución en un momento dado, esto es, los valores de los registros de la CPU para dicho programa.
 - Su memoria de trabajo, es decir, la memoria que ha reservado y sus contenidos.
 - Otra información que permite al sistema operativo su planificación.
- **Bloque de control del proceso (BCP):** en inglés **PCB (Process Control Block)** es un registro especial donde el sistema operativo agrupa toda la información que necesita conocer respecto a un proceso particular. Cada vez que se crea un proceso el sistema operativo crea el BCP correspondiente para que sirva como descripción en tiempo de ejecución durante toda la vida del proceso.

Referencias

- [http://es.wikipedia.org/wiki/Proceso_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Proceso_(inform%C3%A1tica))
- http://es.wikipedia.org/wiki/Bloque_de_control_del_proceso

Especificación de Requerimientos

Requerimientos Funcionales

Los requerimientos funcionales representan todas aquellas funciones básicas y primordiales del módulo. En esta sección se presenta un breve resumen que permitirá entender el requerimiento y hacerle seguimiento a lo largo del desarrollo del sistema se describe la referencia del requerimiento, impacto, descripción, tipo (funcionalidad, confiabilidad, seguridad, desempeño, usabilidad), relación con otros requerimientos o módulos, observaciones y/o soporte:

• <i>Crear proceso</i>	
Referencia	RF.2.2.1 Crear un proceso
Impacto (Riesgo)	Alto
Requerimiento o Descripción	El sistema deberá crear 3 procesos, indicando los valores de la pila y del registro eip además de la sección Tss, para cada uno de ellos
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	Se ejecutaran los procesos indicados, en este caso sólo 2 de ellos, uno que imprimirá la palabra "hola" en pantalla y el otro que imprimirá la palabra "chao".
Comportamiento	
Relaciones	
Observaciones y/o soporte	Los procesos son "infinitos" este comportamiento podría cambiar en el futuro.

Actores

Requerimientos No Funcionales

Los requerimientos no funcionales asociados al módulo en desarrollo, son listados en ésta sección.

Referencia	RC 2.2.1 Cumplimiento de especificaciones.
Requerimiento o Descripción	El Sistema debe cumplir con todas las especificaciones propuestas.
Tipo	
Relaciones	N/A
Observaciones y/o soporte	

Referencia	RC 2.2.2 Posibilidad de Incorporar cambios
Requerimiento o Descripción	El Sistema debe tener la disponibilidad y facilidad de incorporar cambios y nuevas funcionalidades al mismo.
Tipo	
Relaciones	N/A
Observaciones y/o soporte	

Referencia	RC 2.2.3 Legibilidad del código
Requerimiento o Descripción	El código del sistema debe ser legible.
Tipo	
Relaciones	N/A
Observaciones y/o soporte	

Actores.

Luego de un estudio entre las partes interesadas en el desarrollo del sistema se elaboró una lista de los actores que darán uso al sistema, estos se presentan a continuación.

Actores	Descripción
Sistema	Módulo encargado de la creación y manejo de procesos

Luego de identificar los usuarios del sistema se continúa con la lista de los casos de uso, los cuales se obtuvieron por medio de un estudio de todos los procesos y eventos que realizan los usuarios dentro de la aplicación.

Para la presentación de los casos de uso se toma como referencia la notación que se ha estado usando hasta el momento (UML).

Diagrama de Casos de Uso

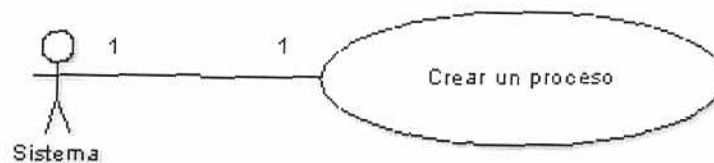


Figura 6 - Casos de Uso. Módulo de Procesos

Elaboración Propia

Vianyx – 2do Entregable

Lista de Cambios

Visión del Módulo Planificador (Mód - 2.3)

Versión 1.0

Número orden	Fecha	Versión	Descripción
1	2018-08-01	1.0	Inicio
2	2018-08-01	1.0	Cambios definiciones
3	2018-08-01	1.0	Cambios del diagrama clases

Lista de Cambios

Número orden	Fecha	Versión	Descripción	Estado	Autor
1	27/07/2008	1.0	inicio	En Revisión	Aneriz Rodríguez
2	01/08/2008	1.0	Cambios definiciones	En Revisión	Víctor Ocanto
3	3/08/2008	1.0	Cambios del diagrama de clases		Víctor Ocanto

Especificación de requerimientos de Software

Tabla de Contenido

Introducción	48
Descripción del Módulo.....	48
Definiciones, Acrónimos y Abreviaciones	48
Referencias	48
Requerimientos Funcionales.....	49
RF.2.3.1 Seleccionar el siguiente proceso a ejecutar	49
RC 2.3.1 Cumplimiento de especificaciones.....	50
RC 2.3.2 Posibilidad de Incorporar cambios	50
Actores.	51
Diagrama de Casos de Uso.....	51

Este módulo se encarga de ejecutar la planificación según la estrategia que se utilice el cual dependerá de la lógica elegida para el planificador. Se le otorgará un determinado tiempo determinado QUANTUM a cada proceso de acuerdo a la estrategia de administración de tiempo de procesador denominada Round Robin, mientras la forma de seleccionar el siguiente proceso será la estrategia FIFO.

El Módulo de Planificador es una parte más importante de la arquitectura de Vianyx, debido a que con las herramientas que éste provee se podrá administrar los procesos y así poder modificar la funcionalidad del sistema.

Definiciones, Acrónimos y Abreviaciones

A continuación se muestran los términos, acrónimos y abreviaciones necesarias para interpretar la funcionalidad de este documento.

- **FIFO (First In First Out)**: Es un método utilizado para el manejo de datos, contabilidad de costos y forma de votar. Se trata de que las personas que esperen en una fila y van en orden, llegando en un momento determinado, es decir, que la primera persona llega es la primera en salir.
- **Round Robin**: Es un método de reparto de recursos distribuido en un grupo de manera equitativa, es decir, que se reparte de manera equitativa por el primer elemento de la lista hasta llegar al final, y luego se reparte desde el primer elemento. El funcionamiento Round Robin de un sistema se refiere a la forma de implementar, y esta forma de implementar.

Referencias

- <http://es.wikipedia.org/wiki/FIFO>
- http://es.wikipedia.org/wiki/Round_Robin

Especificación de requerimientos de Software

Introducción

Este documento proporciona una vista funcional del Módulo del Planificador del 2do entregable de Vianyx, identificando sus funcionalidades principales y complementarias, aspectos de diseño, restricciones y otros aspectos basados en los diagramas y definiciones de los casos de uso.

Descripción del Módulo

El Módulo del Planificador del 2do entregable de Vianyx tiene la finalidad de explicar, definir y dar los detalles más importantes acerca de la planificación de procesos en el sistema, explicar sus definiciones y como debería actuar el sistema.

Este módulo se encarga de ejecutar la planificación según del algoritmo que se utilice el cual dependerá de la lógica elegida para el planificador, en este caso se otorgara un determinado tiempo determinado QUANTUM a cada proceso, formando así la estrategia de administración de tiempo de procesador denominada Round Robin, mientras la forma de seleccionar el siguiente proceso será la estrategia FIFO.

El Módulo del Planificador es una parte más importante del 2do entregable de Vianyx, debido a que con las funcionalidades que éste provee, pueden planificarse los procesos y así poder mostrar la funcionalidad del sistema.

Definiciones, Acrónimos y Abreviaciones

A continuación se describen los términos, acrónimos, y abreviaciones necesarias para interpretar de forma apropiada el presente documento.

- **FIFO *First In First Out*:** Es un método utilizado en estructuras de datos, contabilidad de costes y teoría de colas. Guarda analogía con las personas que esperan en una cola y van siendo atendidas en el orden en que llegaron, es decir, que *la primera persona que entra es la primera persona que sale*.
- **Round Robin:** Es un método para seleccionar todos los elementos en un grupo de manera equitativa y en un orden racional, normalmente comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento. El planeamiento Round Robin es tan simple como fácil de implementar, y está libre de inanición.

Referencias

- <http://es.wikipedia.org/wiki/FIFO>
- http://es.wikipedia.org/wiki/Planificaci%C3%B3n_Round-robin

Requerimientos • Especificación de Requerimientos.

Los requerimientos no se asocian directamente al módulo.

Requerimientos Funcionales

Los requerimientos funcionales representan todas aquellas funciones básicas y primordiales del módulo. En esta sección se presenta un breve resumen que permitirá entender el requerimiento y hacerle seguimiento a lo largo del desarrollo del sistema se describe la referencia del requerimiento, impacto, descripción, tipo (funcionalidad, confiabilidad, seguridad, desempeño, usabilidad), relación con otros requerimientos o módulos, observaciones y/o soporte:

• <i>Seleccionar el siguiente proceso a ejecutar</i>	
Referencia	RF.2.3.1 Seleccionar el siguiente proceso a ejecutar
Impacto (Riesgo)	Alto
Requerimiento o Descripción	El sistema podrá seleccionar, basado en la estrategia determinada (FIFO), el siguiente proceso a ejecutar, y dicha planificación se activará en el un intervalo regular de tiempo.
Tipo	Funcionalidad
Base de Datos	Ninguna
Criterios de Aceptación	• Se ejecutaran procesos de forma alternante.
Comportamiento	
Relaciones	Módulo de Reloj
Observaciones y/o soporte	• Se espera que la implementación de este módulo cambie a futuro.

Requerimientos No Funcionales

Los requerimientos no funcionales asociados al módulo en desarrollo, son listados en ésta sección.

Referencia	RC 2.3.1 Cumplimiento de especificaciones.
Requerimiento o Descripción	El Sistema debe cumplir con todas las especificaciones propuestas.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 2.3.2 Posibilidad de Incorporar cambios
Requerimiento o Descripción	El Sistema debe tener la disponibilidad y facilidad de incorporar cambios y nuevas funcionalidades al mismo.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 2.3.3 Legibilidad del código
Requerimiento o Descripción	El código del sistema debe ser legible.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Actores.

Luego de un estudio entre las partes interesadas en el desarrollo del sistema se elaboró una lista de los actores que darán uso al sistema, estos se presentan a continuación.

Actores	Descripción
Sistema	Módulo encargado de ejecutar la política de planificación

Luego de identificar los usuarios del sistema se continúa con la lista de los casos de uso, los cuales se obtuvieron por medio de un estudio de todos los procesos y eventos que realizan los usuarios dentro de la aplicación.

Para la presentación de los casos de uso se toma como referencia la notación que se ha estado usando hasta el momento (UML).

Diagrama de Casos de Uso

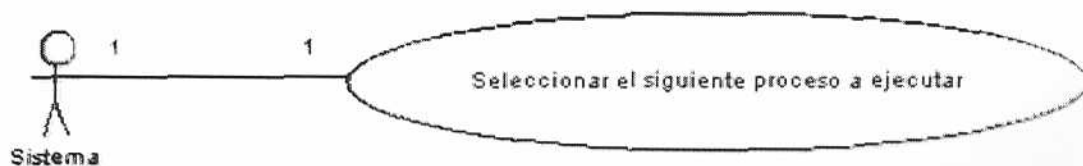


Figura 7 - Casos de Uso. Módulo del Planificador

Elaboración Propia.

Vianyx – 2do Entregable

Lista de Cambios

Visión del Módulo Consola (Mód - 2.4)

Versión 1.0

Número orden	Fecha	Versión	Descripción	Responsable
1	2018-08-20	1.0	Inicio	Agenda
2	2018-08-20	1.0	Cambios en los casos de uso	Victor Ochoa
3	2018-08-20	1.0	Cambios del diagrama de clases y en las definiciones	Victor Ochoa

Lista de Cambios

Número orden	Fecha	Versión	Descripción	Estado	Autor
1	05/08/2008	1.0	inicio	En Revisión	Aneriz Rodríguez
2	7/08/2008	1.0	Cambios en los casos de uso	En Revisión	Víctor Ocanto
3	8/08/2008	1.0	Cambios del diagrama de clases y en las definiciones.	En Revisión	Víctor Ocanto

Tabla de Contenido

Introducción	55
Descripción del Módulo.....	55
Definiciones, Acrónimos y Abreviaciones	55
Referencias	55
Requerimientos Funcionales.....	56
RF.2.4.1 Interpretar los comandos	56
RC 2.4.1 Cumplimiento de especificaciones.....	57
RC 2.4.2 Posibilidad de Incorporar cambios	57
Actores.....	58
Diagrama de Casos de Uso.....	58

Especificación de Requerimientos

Introducción

Este documento proporciona una vista funcional del Módulo de Consola del 2do entregable de Vianyx, identificando sus funcionalidades principales y complementarias, aspectos de diseño, restricciones y otros aspectos basados en los diagramas y definiciones de los casos de uso.

Descripción del Módulo

El Módulo de Consola del 2do entregable de Vianyx tiene la finalidad de explicar, definir y dar los detalles más importantes acerca de la implementación de la consola en el sistema, explicar sus definiciones y como debería actuar el sistema.

Este módulo se encarga de interpretar ciertas instrucciones que el usuario seleccione, la interfaz será en modo texto y estará basada en menús en donde se presentaran diversas opciones que el usuario podrá elegir.

El Módulo de Consola es una parte fundamental dentro del 2do entregable de Vianyx, debido a las funcionalidades que éste provee, se considera como la parte final del desarrollo e integra todos los demás módulos, para proveer un sistema funcional.

Definiciones, Acrónimos y Abreviaciones

A continuación se describen los términos, acrónimos, y abreviaciones necesarias para interpretar de forma apropiada el presente documento.

- **Modo Texto:** Se llama modo texto a cualquier modo de vídeo de un ordenador en el que el contenido de la pantalla se representa internamente en términos de caracteres textuales en lugar de píxeles individuales. Típicamente, la pantalla consistirá en una rejilla uniforme de «celdas de caracteres», cada una de las cuales contendrá un carácter del juego de caracteres.

Referencias

- http://es.wikipedia.org/wiki/Modo_texto

Requisitos • Especificación de Requerimientos.

Los requerimientos no funcionales asociados al módulo en desarrollo, son listados en

Requerimientos Funcionales

Los requerimientos funcionales representan todas aquellas funciones básicas y primordiales del módulo. En esta sección se presenta un breve resumen que permitirá entender el requerimiento y hacerle seguimiento a lo largo del desarrollo del sistema se describe la referencia del requerimiento, impacto, descripción, tipo (funcionalidad, confiabilidad, seguridad, desempeño, usabilidad), relación con otros requerimientos o módulos, observaciones y/o soporte:

• Interpretar los comandos	
Referencia	RF.2.4.1 Interpretar los comandos
Impacto (Riesgo)	Alto
Requerimiento o Descripción	Se tendrá una serie de opciones, de la cual se puede seleccionar una, el sistema deberá realizar el comportamiento establecido.
Tipo	
Base de Datos	
Criterios de Aceptación	N/A El sistema deberá responder a los comandos elegidos por el usuario, en este caso se presentaran 3 comandos básicos los cuales son: a) Mostrar la tabla de procesos b) Activar el planificador c) Desactivar el planificador
Comportamiento	Ante la opción seleccionada el sistema deberá realizar el comportamiento establecido, si la selección no se encuentra en el menú se ignora dicha opción.
Relaciones	RF.2.4.2
Observaciones y/o soporte	Se espera que los comandos sean ampliados , en el futuro

Requerimientos No Funcionales

Los requerimientos no funcionales asociados al módulo en desarrollo, son listados en ésta sección.

Referencia	RC 2.4.1 Cumplimiento de especificaciones.
Requerimiento o Descripción	El Sistema debe cumplir con todas las especificaciones propuestas.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 2.4.2 Posibilidad de Incorporar cambios
Requerimiento o Descripción	El Sistema debe tener la disponibilidad y facilidad de incorporar cambios y nuevas funcionalidades al mismo.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Referencia	RC 2.4.3 Legibilidad del código
Requerimiento o Descripción	El código del sistema debe ser legible.
Tipo	
Relaciones	N /A
Observaciones y/o soporte	

Actores.

Luego de un estudio entre las partes interesadas en el desarrollo del sistema se elaboró una lista de los actores que darán uso al sistema, estos se presentan a continuación.

Actores	Descripción
Sistema	Módulo de teclado, encargado de interpretar los comandos introducidos por el usuario

Luego de identificar los usuarios del sistema se continúa con la lista de los casos de uso, los cuales se obtuvieron por medio de un estudio de todos los procesos y eventos que realizan los usuarios dentro de la aplicación.

Para la presentación de los casos de uso se toma como referencia la notación que se ha estado usando hasta el momento (UML).

Diagrama de Casos de Uso

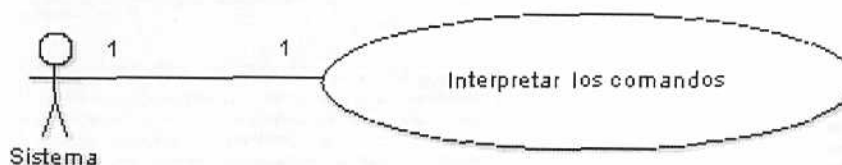


Figura 8 - Casos de Uso. Módulo de Consola
Elaboración Propia.

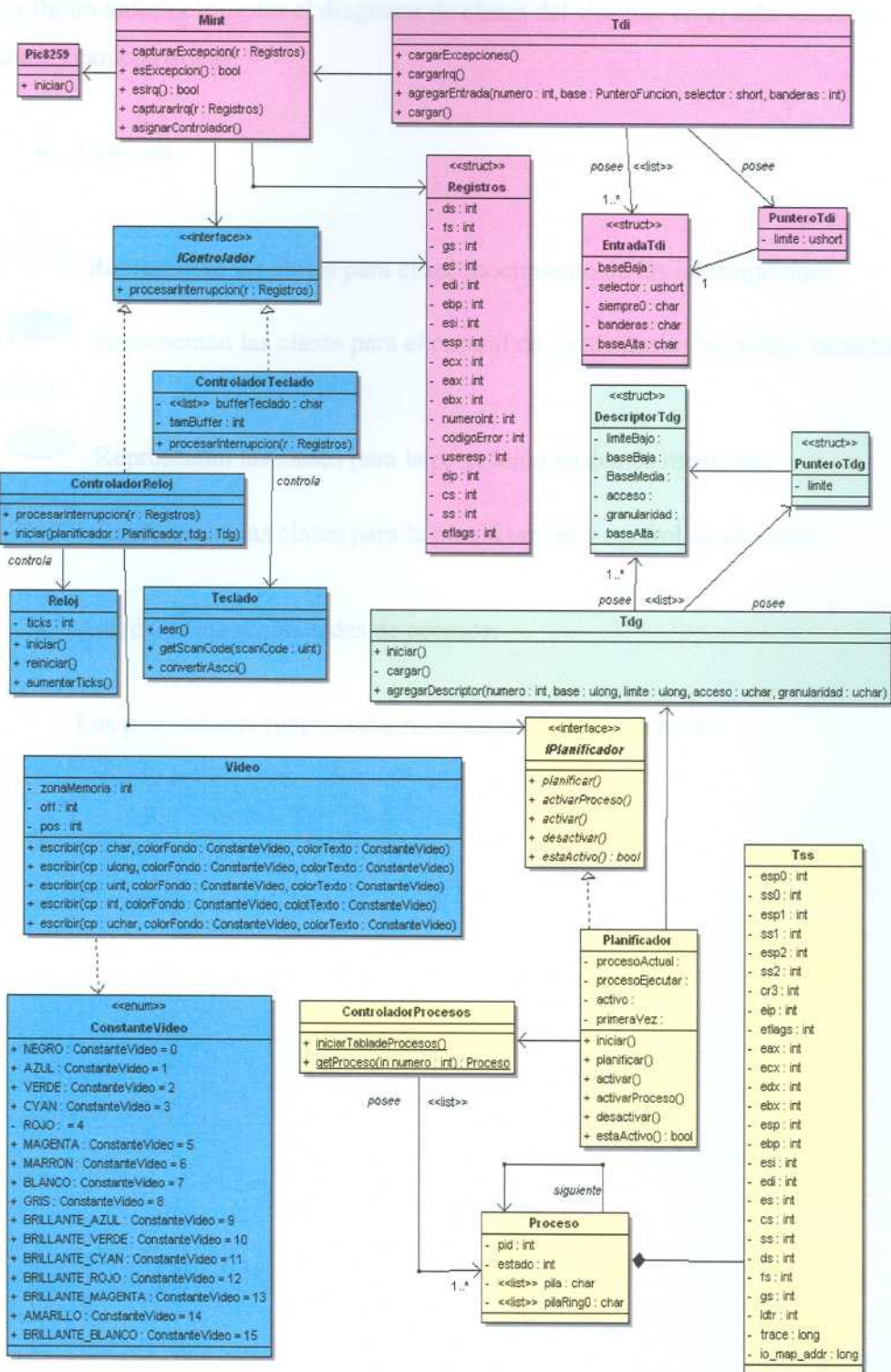


Figura 9 - Diagrama de clases del sistema

Fuente: Creación Propia

La figura anterior muestra el diagrama de clases del sistema, en el cabe destacar ciertas características:

- Leyenda:



Representan las clases para el reconocimiento de las interrupciones.



Representan las clases para el control de los dispositivos (video, teclado, reloj).



Representan las clases para la protección básica de memoria.



Representan las clases para la planificación y control de procesos.

- Acerca de las propiedades de accesos:

Las propiedades (get*, set*), no se encuentran modeladas.

APÉNDICE B – CASOS DE PRUEBA

Caso de Prueba Módulo Arranque

El caso de prueba para el módulo de arranque consiste en verificar que las instrucciones de inicialización del sistema, las cuales se ejecutan al iniciar el sistema, se ejecuten correctamente y se guarden en el archivo de configuración. Los datos de prueba se obtienen de la imagen de arranque. En la imagen de arranque se muestra la pantalla de inicio de sesión, la cual se ejecuta sobre el sistema de arranque.

Apéndice B

Casos de Prueba



La figura muestra la pantalla de inicio de sesión, la cual se ejecuta sobre el sistema de arranque. En ella se puede observar el sistema de arranque y posteriormente se ejecuta el módulo de arranque.

APÉNDICE B – CASOS DE PRUEBA

Caso de Prueba Módulo Arranque

La prueba sobre el módulo de arranque consistió en agregar unas instrucciones básicas al núcleo inicial del sistema, las cuales establecerían los valores de los registros `ebx` y `ecx` a `0x10`, antes de terminar la ejecución, y en el archivo de salida `bochs.log` dichos registros reflejarían el valor indicado. En la imagen siguiente se muestra la pantalla inicial que presenta GRUB corriendo sobre el simulador Bochs.

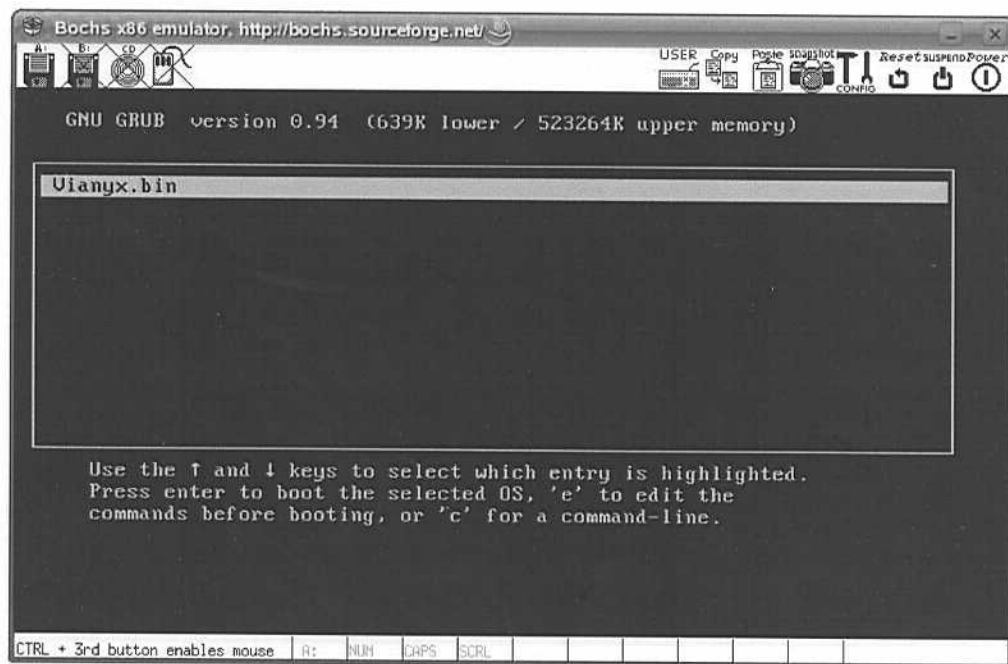


Figura 10 - Pantalla inicial mostrada por el gestor de arranque.

Fuente: Elaboración propia.

La figura anterior muestra la interfaz presentada por GRUB, en ella se puede observar el nombre del núcleo que puede cargarse en memoria y posteriormente

ser ejecutado, en este caso la opción Vianyx.bin representa el núcleo básico que contiene las instrucciones para cambiar el valor de los registros, como se indicó anteriormente.

Al seleccionar la opción deseada, el gestor colocará en memoria el código del núcleo y pasará el control al mismo, GRUB imprimirá unos mensajes indicando: el dispositivo donde se encontraba almacenado el ejecutable del núcleo, el tipo de sistema de archivos que posee el dispositivo, la línea que describe la ruta relativa del archivo y el formato ejecutable en el que se encuentra el núcleo. En este caso el formato es ELF¹, todos estos datos se muestran a continuación.

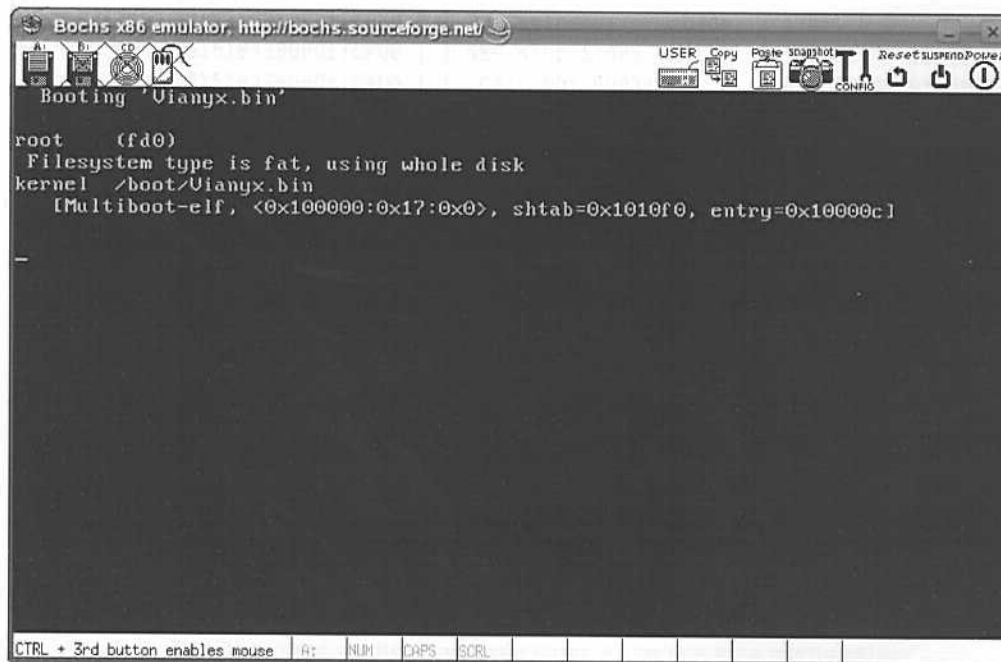


Figura 11 - Arranque del sistema.

Fuente: Elaboración propia.

¹ El formato **ELF** (*Executable and Linkable Format*) es un formato de archivo para ejecutables, código objeto, librerías compartidas y volcados de memoria. Fue desarrollado por el UNIX System Laboratory (USL) como parte de la ABI. En principio fue desarrollado para plataformas de 32 bits, a pesar de que hoy en día se usa en gran variedad de sistemas.

La ejecución del núcleo no tiene efecto visible alguno, ya que en esta etapa aún no se contaba con los módulos necesarios para generar información en la pantalla, sin embargo el simulador genera un archivo con la información del estado de los registros, entre otros. Dicho archivo se muestra a continuación:

```

bochsout.txt Kate
Archivo Editar Proyecto Documento Ver Marcadores Herramientas Preferencias Ventana Ayuda

51518738000i[CPU0 ] CS.d_b = 16 bit
51518738000i[CPU0 ] SS.d_b = 16 bit
51518738000i[CPU0 ] | EAX=00000000 EBX=00000010 ECX=00000010
EDX=00000000
51518738000i[CPU0 ] | ESP=00001ff4 EBP=00001ff0 ESI=0002cb87
EDI=0002cb88
51518738000i[CPU0 ] | IOPL=0 id vip vif ac vm rf nt of df IF tf sf zf
af PF cf
51518738000i[CPU0 ] | SEG selector      base      limit G D
51518738000i[CPU0 ] | SEG sltr(index|ti|rpl)      base      limit G D
51518738000i[CPU0 ] | CS:0000( 0003| 0| 0) 00000000 0000ffff 0 0
51518738000i[CPU0 ] | DS:0000( 0004| 0| 0) 00000000 0000ffff 0 0
51518738000i[CPU0 ] | SS:0000( 0004| 0| 0) 00000000 0000ffff 0 0
51518738000i[CPU0 ] | ES:0000( 0004| 0| 0) 00000000 0000ffff 0 0
51518738000i[CPU0 ] | FS:0000( 0004| 0| 0) 00000000 0000ffff 0 0
51518738000i[CPU0 ] | GS:0000( 0004| 0| 0) 00000000 0000ffff 0 0
51518738000i[CPU0 ] | EIP=000082b6 (000082b6)
51518738000i[CPU0 ] | CR0=0x00000010 CR1=0 CR2=0x00000000
51518738000i[CPU0 ] | CR3=0x00000000 CR4=0x00000000
51518738000i[CPU0 ] >> jmp .+0xffff (0x000082b3) : E9FAFF
51518738000i[      ] restoring default signal behavior
51518738000i[CTRL ] quit_sim called with exit code 1

Line: 1 col: 1  INS NORM bochsout.txt

Bochs is exiting with the following message:
[XBUT] POWER button turned off.

victorellimux:~/Desktop/testis/Vianyx-Arranque> cd '/home/victor/Desktop/testis/Vianyx-Arranque'
victorellimux:~/Desktop/testis/Vianyx-Arranque> cd '/home/victor/Desktop/testis/Vianyx-Arranque/nucleo/Arranque'
victorellimux:~/Desktop/testis/Vianyx-Arranque/nucleo/Arranque> cd '/home/victor/Desktop/testis/Vianyx-Arranque'
victorellimux:~/Desktop/testis/Vianyx-Arranque> cd '/home/victor/Desktop/testis/Vianyx-Arranque/nucleo'
victorellimux:~/Desktop/testis/Vianyx-Arranque/nucleo> cd '/home/victor/Desktop/testis/Vianyx-Arranque'
victorellimux:~/Desktop/testis/Vianyx-Arranque>

Encontrar en archivos Terminal
  
```

Figura 12 - Archivo de logs de Bochs

Fuente: Elaboración propia.

En la figura anterior se aprecia el estado de los registros (en el marco de color rojo). Como puede observarse, los registros `eax` y `ebx` poseen el valor `0x10`, lo que indica el éxito de la ejecución.

El código que establece los valores de los registros en el núcleo se muestra a continuación:

```
_start:
    mov ebx, 0x10
    mov ecx, 0x10
    ret ;
```

Figura 13 - Código inicial de la función `Start`.

Fuente: Elaboración propia.

Caso de Prueba Video

Se utilizó una prueba unitaria para este módulo donde cada método de la clase que representa la prueba se encarga de comprobar el correcto funcionamiento de las diferentes funciones que ofrece el módulo de video.

La ejecución de la prueba unitaria se muestra a continuación:



Figura 14 - Ejecución de la prueba unitaria para el video

Fuente: Elaboración propia.

En la figura anterior se observan diferentes mensajes, donde cada uno de ellos indica que se está realizando una prueba e indican el valor o comportamiento esperado, las funcionalidades probadas para este módulo fueron:

- Capacidad de mostrar un mensaje indicando colores de fondo y texto.
- Capacidad de mostrar distintos tipos de datos entre estos :
Se probaron las respuestas del sistema ante los siguientes casos:
 - unsigned long: representa un entero largo sin signo.
 - unsigned int: representa un entero simple sin signo.
 - int: representa un entero simple con signo.
 - char: tipo de dato que representa un carácter imprimible en pantalla

Al observar el comportamiento del sistema durante la ejecución de la prueba, se concluyó que el módulo de video cumplía con lo esperado.

Caso de prueba Interrupciones

Se probaron las respuestas del sistema ante los siguientes casos:

- a) Excepciones por fallo al dividir por cero
- b) Excepción por intento de acceder a un segmento de memoria no presente.
- c) Excepción por intento de acceder al segmento de memoria nulo.
- d) Excepción por intento de acceder a una posición de memoria fuera del segmento establecido.

Las imágenes a continuación muestran la ejecución de cada una de las pruebas realizadas:

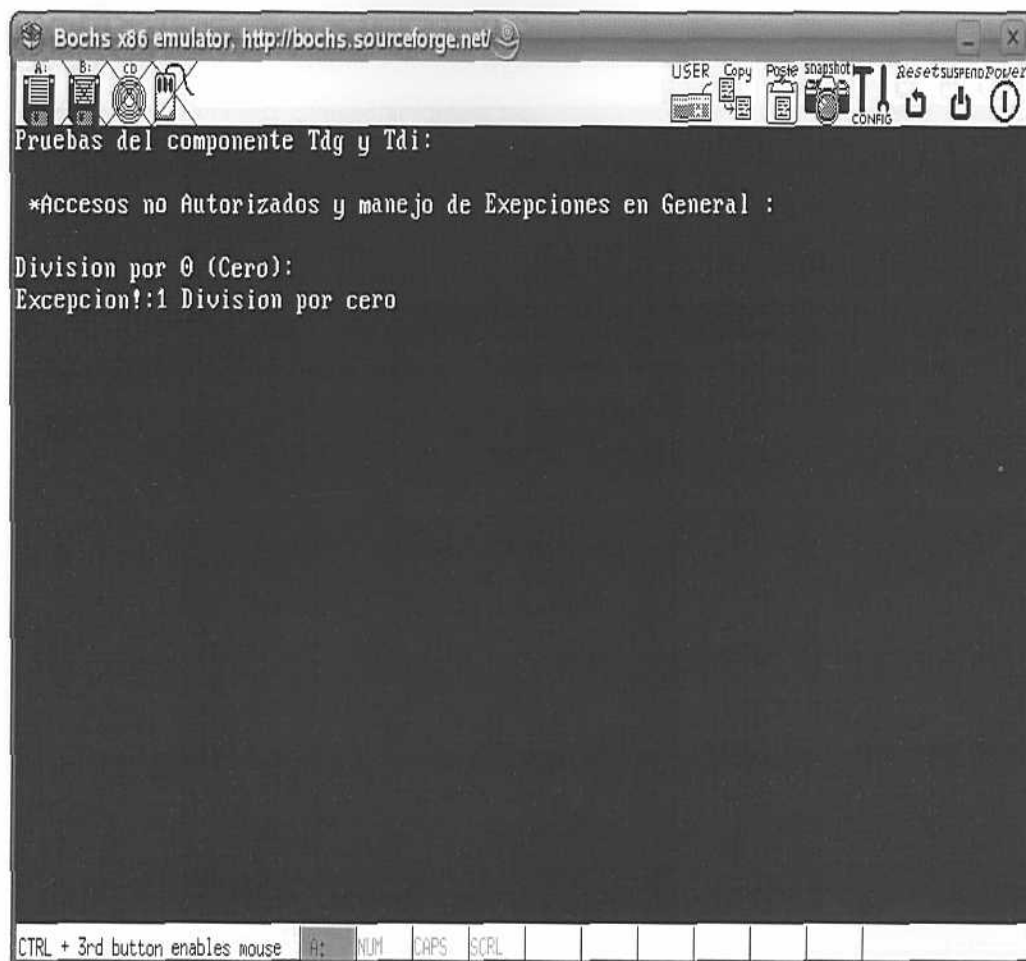


Figura 15 - Ejecución de la prueba de división entre cero

Fuente: Elaboración propia.

La imagen anterior muestra la ejecución de la primera prueba, para ello se creó un método que, de forma deliberada, realiza una división entre cero, (el compilador de C++ sólo genera un aviso), la ejecución de este método produce una excepción que será capturada por el sistema, luego este mostrará un mensaje indicando que la excepción fue reconocida y capturada, y así el sistema detendrá su ejecución.

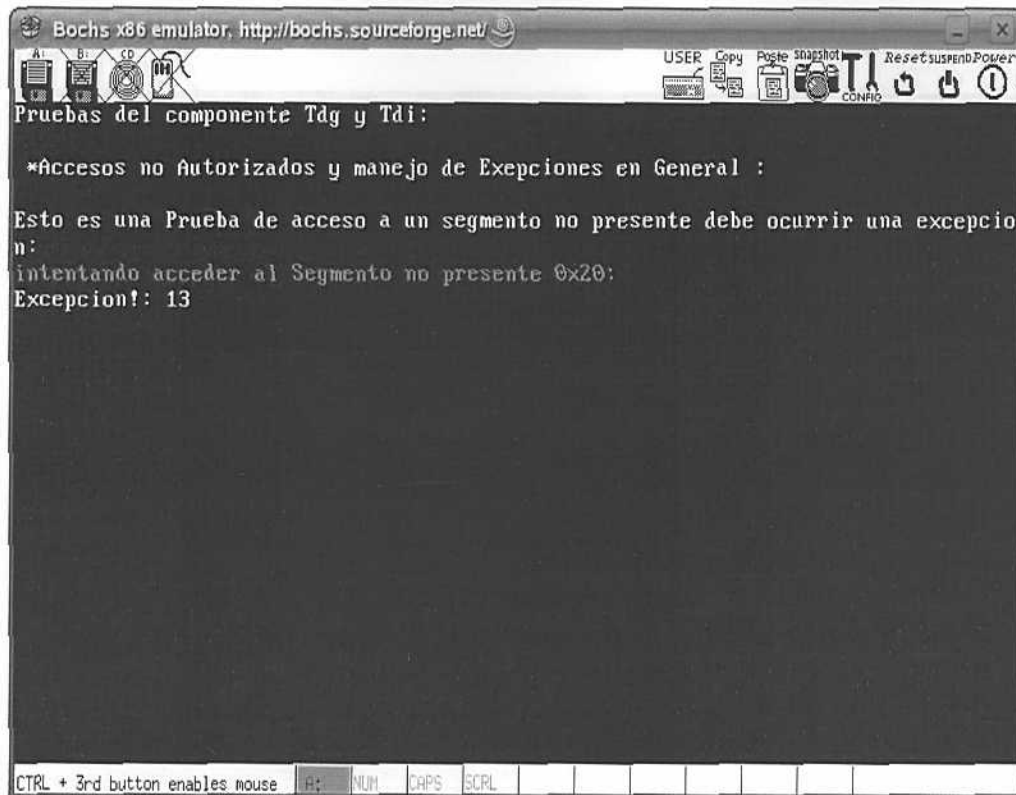


Figura 16 – Ejecución de la prueba de acceso a un segmento no presente.

Fuente: Elaboración propia.

La imagen anterior muestra la ejecución de la segunda prueba, en el método setUp de esta prueba se establece una Tdg (Tabla de descriptores globales) con tres segmentos, cuyos valores de acceso son 0x0 segmento nulo, 0x10 segmento de código y 0x18 segmento de datos. El método de prueba intenta realizar el acceso a un segmento no definido en la Tdg, en este caso el valor de acceso es 0x20, lo cual producirá una excepción.

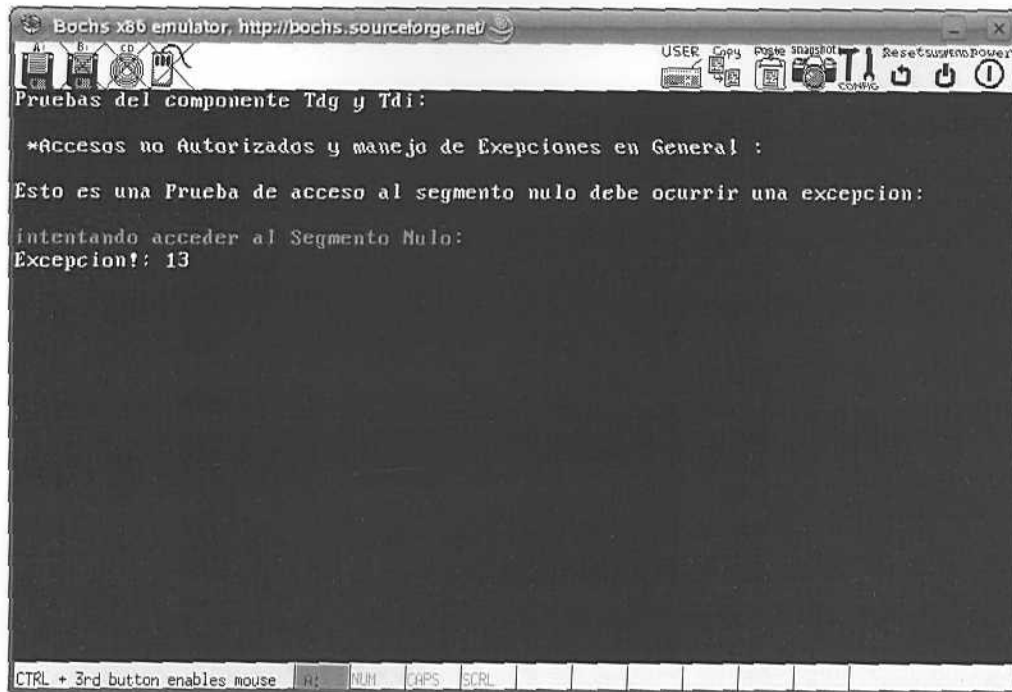


Figura 17 – Ejecución de la prueba intentando acceder a un segmento nulo.

Fuente: Elaboración propia.

La figura anterior muestra los resultados de la ejecución del método de prueba intento de acceso al segmento nulo.

El segmento nulo debe ser definido de forma obligatoria, pero cualquier intento de acceso a este segmento por medio del selector 0x0, producirá una excepción. Por ello, se programó un método que intentara realizar un acceso a dicho segmento para la prueba.

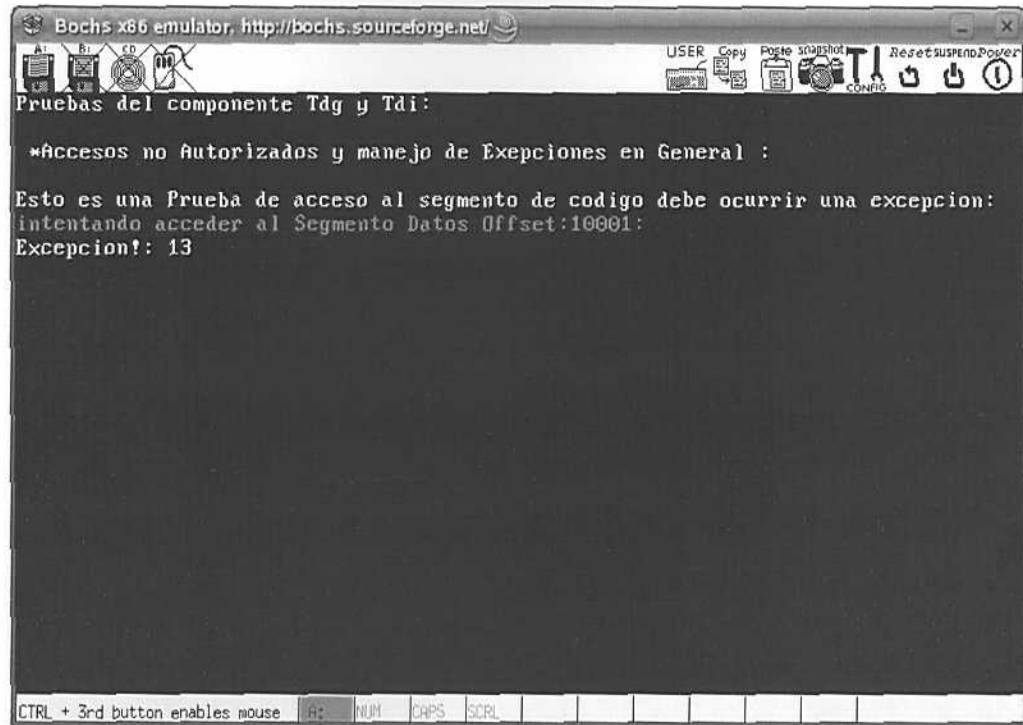


Figura 18 - Ejecución de la prueba intentando acceder al segmento de datos.

Fuente: Elaboración propia.

La figura anterior muestra los resultados de la ejecución del método de prueba, que intenta sobrepasar el límite establecido en el segmento (1000 bytes), en este caso se intenta referenciar un dato que se encuentra en la dirección 1001, lo cual produce una excepción.

Caso de Prueba Módulo Teclado

Para este caso de prueba se instanció un controlador de teclado y se observó la respuesta del sistema al escribir alguna frase.

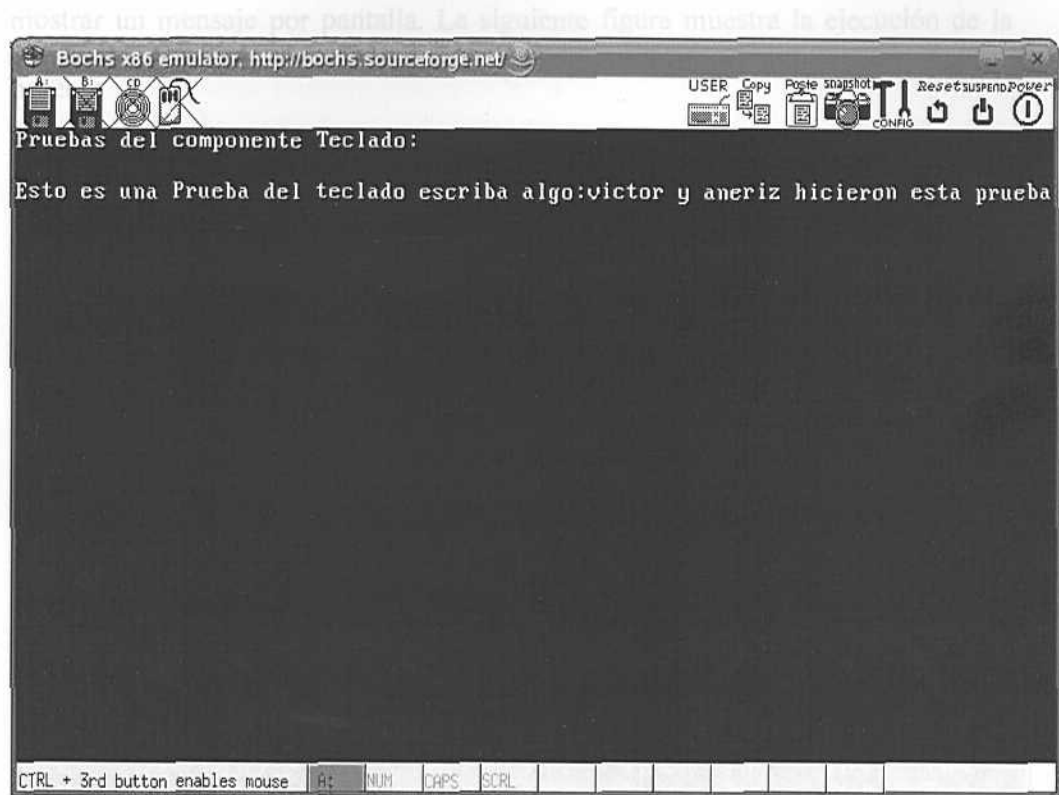


Figura 19 - Prueba del módulo de teclado

Fuente: Elaboración propia.

La figura anterior muestra la ejecución de la prueba para este módulo. Se esperaba en esta prueba que al pulsar los caracteres imprimibles del teclado, estos fueran mostrados en la pantalla del computador.

Caso de Prueba Módulo Reloj

El reloj genera una interrupción a intervalos regulares de tiempo (dependiendo del procesador), la prueba para este módulo consistió en capturar la interrupción y mostrar un mensaje por pantalla. La siguiente figura muestra la ejecución de la prueba:

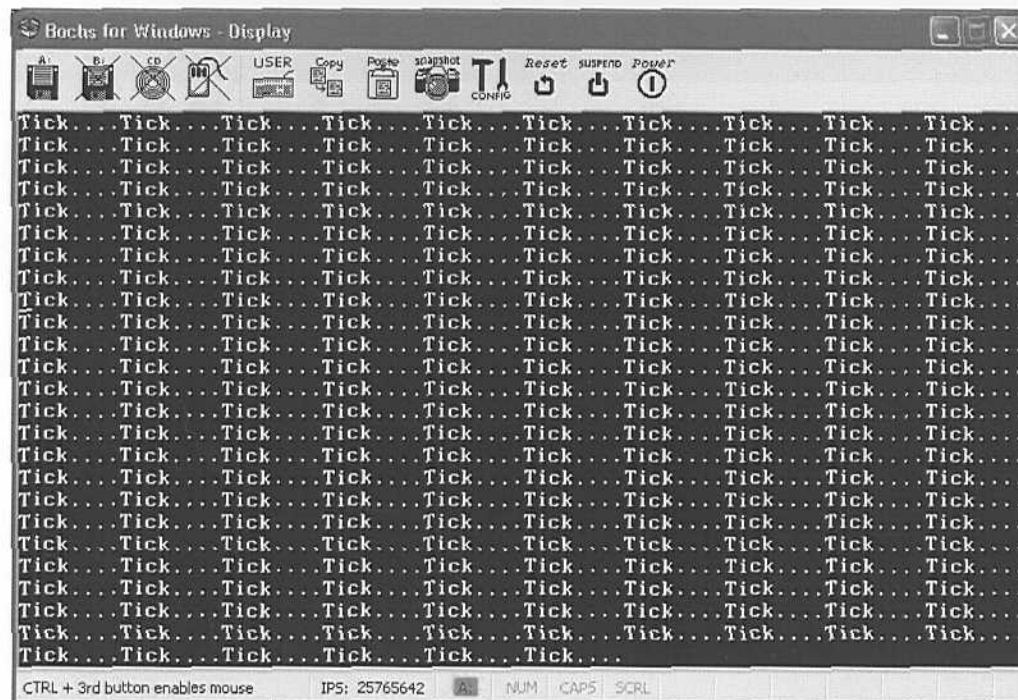


Figura 20 - Ejecución de la prueba del módulo de reloj.

Fuente: Elaboración propia.

Como se puede observar en la figura, durante la ejecución de la prueba se imprimía la frase "Tick..." repetidamente, indicando que se logró capturar la interrupción de reloj.

Caso de Prueba Módulo Planificador

El planificador, según la estrategia de planificación que se haya programado, en este caso FIFO y Round Robin, ejecutaría un proceso (Proceso A), para luego en un determinado instante indicado por la constante QUANTUM, interrumpirlo y ejecutar el segundo proceso (Proceso B). Las siguientes imágenes muestran la ejecución de la prueba para este módulo.

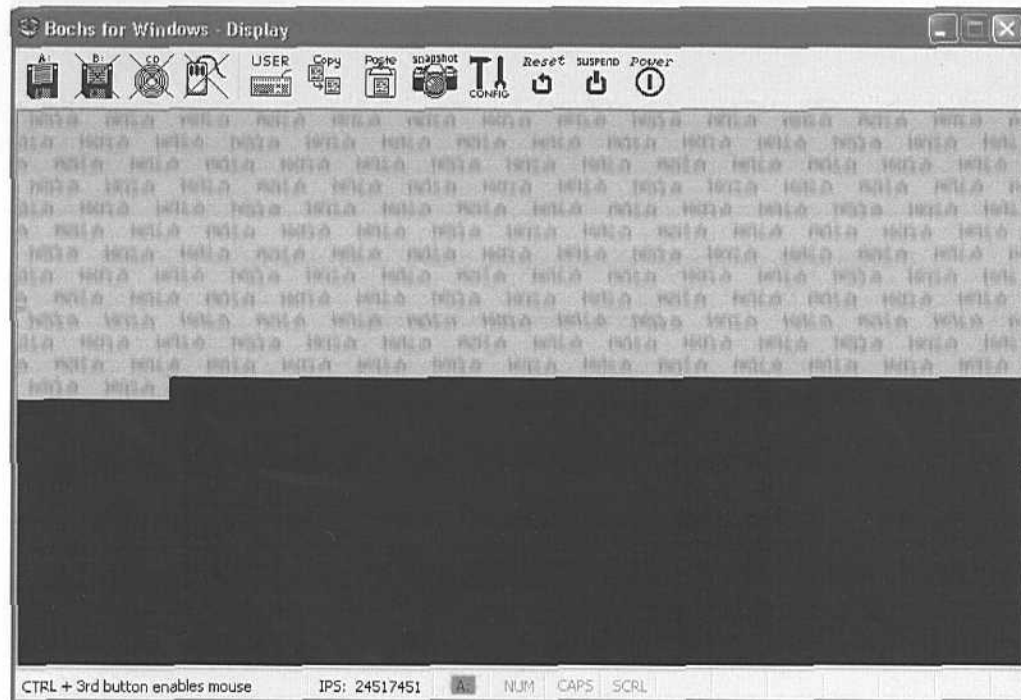


Figura 21 - Ejecución de la prueba del planificador (ejecución del proceso A).

Fuente: Elaboración propia.

En la imagen anterior se observa la ejecución del Proceso A, el cual imprime en pantalla el mensaje “HOLA” en color verde brillante sobre un fondo blanco. En este instante el QUAMTUN de dicho proceso no ha expirado por lo que no es interrumpido.

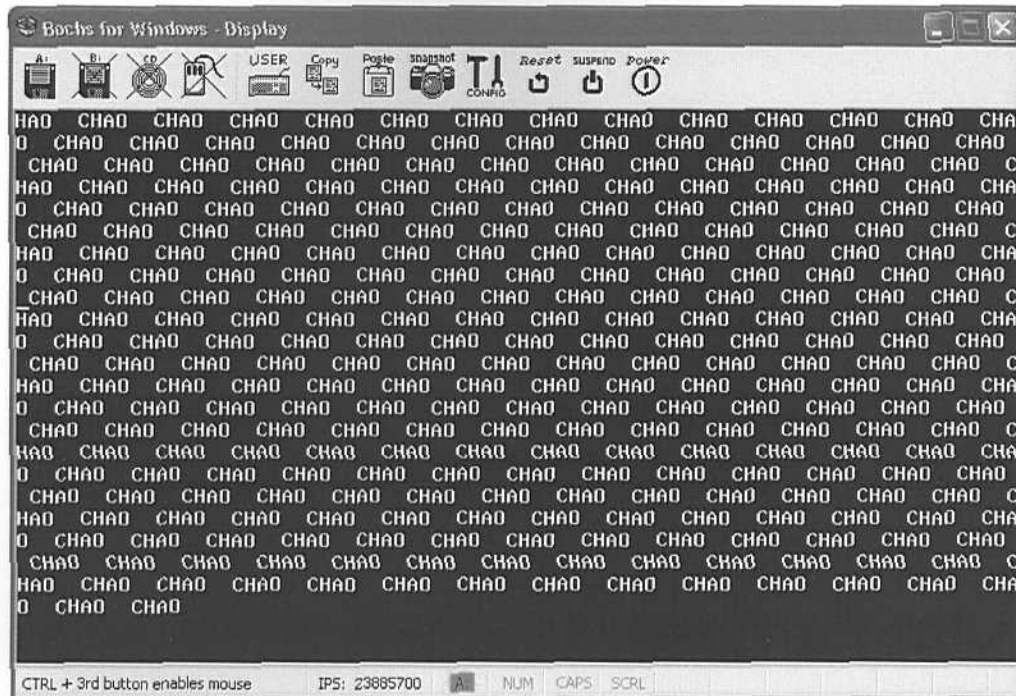


Figura 22 - Ejecución de la prueba del planificador (ejecución del proceso B).

Fuente: Elaboración propia.

En este caso la figura muestra la ejecución del Proceso B, este proceso muestra el mensaje “CHAO” en color blanco brillante sobre un fondo negro. Al igual que el caso anterior el proceso no es interrumpido.

Proceso B

El planificador lee el tiempo de ejecución de cada proceso y lo compara con el tiempo de ejecución de los procesos que están en la cola de ejecución.

considera que el proceso que tiene el menor tiempo de ejecución es el que debe ejecutarse.

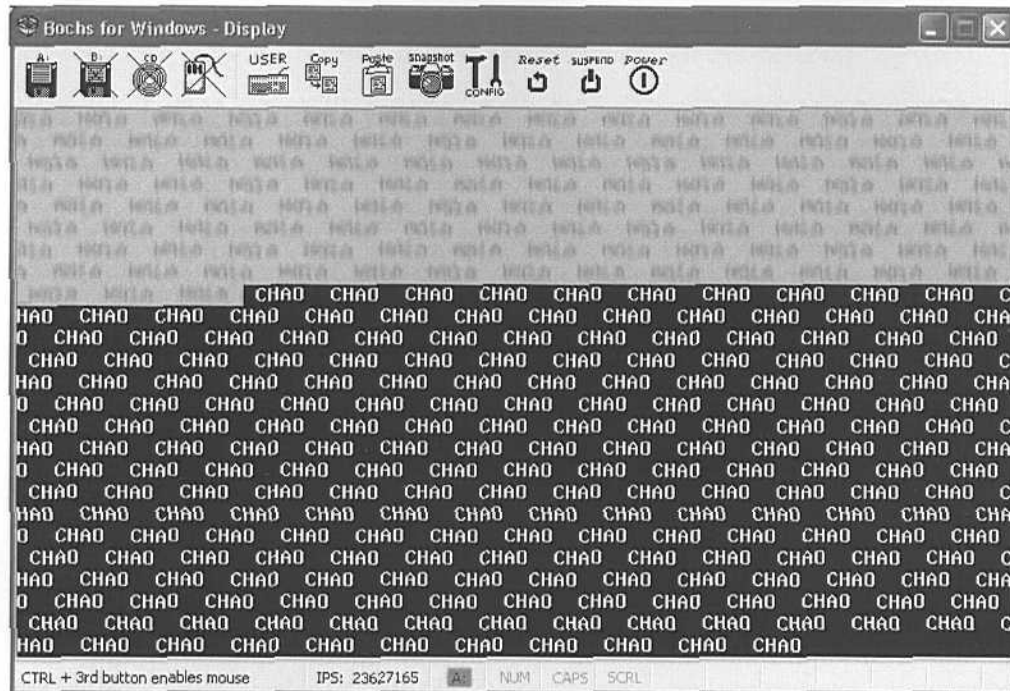


Figura 23 - Ejecución de la prueba del planificador (ejecución de ambos procesos).

Fuente: Elaboración propia.

En esta figura se muestra el momento en el que ocurre el cambio de proceso, el primero de los procesos (Proceso A), se encuentra en ejecución, su tiempo de procesador termina y el planificador elige el siguiente proceso a ejecutar (Proceso B), por lo tanto el Proceso A es interrumpido y comienza la ejecución del Proceso B.

El planificador tuvo el comportamiento esperado en esta prueba por lo que se consideró que el módulo cumplía con los requerimientos.

PRÁCTICA MÓDULO VIDEO

PRÁCTICA MÓDULO VIDEO

El propósito de esta práctica es que el alumno extienda al funcionamiento del controlador de video la funcionalidad que permita indicar la posición donde se desea que el controlador muestre una pantalla (esta posición será otorgada de forma matricial).

Apéndice C

PRÁCTICA MÓDULO VIDEO

PRÁCTICA MÓDULO VIDEO

Prácticas

El controlador VGA en modo texto tiene una zona de memoria con un tamaño de 80x25 y que puede verse como un mapa de bits. Este mapa de bits puede ser manipulado de la siguiente manera:

1. Se define una zona de memoria.

2. Se define una zona de memoria.

3. Se define una zona de memoria.

4. Se define una zona de memoria.

5. Se define una zona de memoria.

6. Se define una zona de memoria.

7. Se define una zona de memoria.

PRÁCTICA MÓDULO VIDEO

OBJETIVO DE LA PRÁCTICA

El objetivo de esta práctica es que el alumno extienda el funcionamiento del controlador de video para agregar una funcionalidad que permita indicar la posición donde se desea que el carácter sea impreso en la pantalla (esta posición será otorgada de forma matricial).

PROBLEMA

EL MÓDULO VIDEO

El controlador VGA en modo texto tiene una zona de memoria que comienza en la dirección 0xB8000 y que puede verse como un arreglo de enteros cortos sin signo (unsigned short), para colocar un carácter en la pantalla basta con colocarlo en una posición de ese arreglo por ejemplo:

```
zonaMemoria = (unsigned short *) 0xB800;
```

```
zonaMemoria[0] = 'H', colocaría la letra 'H' en la pantalla.
```

Actualmente el controlador no permite indicar la posición de impresión por pantalla, en lugar de esto las variables privadas pos y off, son las que indican el lugar de la pantalla donde se debe colocar el carácter y estas variables son incrementadas automáticamente por el controlador, además no usan las coordenadas matriciales sino un algoritmo particular que obtiene el lugar en *RAM* donde debe colocarse el carácter.

DESCRIPCIÓN DE LA FUNCIONALIDAD PEDIDA

Se desea que el estudiante sobrecargue la función *escribir* del módulo de video de forma que se permita indicar la posición del carácter que se desea imprimir.

Ejemplo:

PANTALLA

	0	1	2	3	80
0	H	O	L	A	
1					
⋮					
25					

En este caso la letra “H” ocupa la coordenada (0,0), la función sería:

`escribir('H', 0, 0, NEGRO, BLANCO);`

La “O” ocupa la coordenada (0,1), la función sería:

`escribir('O', 0, 1, NEGRO, BLANCO);`

La “L” ocupa la coordenada (0,2), la función sería:

`escribir('L', 0, 2, NEGRO, BLANCO);`

La “A” ocupa la coordenada (0,3), la función sería:

`escribir('A', 0, 3, NEGRO, BLANCO);`

CARACTERÍSTICAS DE LA FUNCIÓN

Valor de retorno: void.

Nombre: escribir

Parametros: int posX, int posY, int colorFondo, int colorTexto

Acceso: public

```
void escribir(char c, int posX, int posY, int colorFondo, int colorTexto);
```

Para realizar la práctica se deben modificar los archivos ubicados en:
/nucleo/Controladores

Video.h

Video.cpp

OTROS REQUERIMIENTOS

posX debe ser menor(estricto) a 80, posY debe ser menor(estricto) a 25

Esto se debe validar (imprimiendo un mensaje que diga valor "X" o "Y" fuera de rango)

CONCEPTOS UTILIZADOS EN LA PRÁCTICA

- Controlador
- VGA
- Manejo de Arreglos
- Sobrecarga de funciones

EL ENTORNO DE DESARROLLO

El entorno de desarrollo de la práctica será Linux.

SOLUCIÓN PROPUESTA

```
void escribir(char c, int posX, int posY, int colorFondo, int colorTexto)
{
    if((posX<80) && (posY<25))
    {
        pos = posX;
        off = posY*80; //ancho de la pantalla
        escribir(c, colorFondo, colorTexto);
    }
    else
    {
        escribir("Valores fuera del rango");
    }
}
```

DESCRIPCIÓN PRÁCTICA MÓDULO PLANIFICADOR

OBJETIVO DE LA PRÁCTICA

El objetivo de esta práctica es que el alumno extienda el funcionamiento del planificador para agregar una funcionalidad donde el criterio de selección sea uno basado en: - Turnos -, para ello se cuentan con dos esquemas:

- Desarrollo desde cero de un planificador que implemente la interfaz IPlanificador.
- Heredar del planificador existente y sobrescribir el método planificar (recomendado)

PROBLEMA

EL MÓDULO PLANIFICACIÓN

El actual modulo de planificación, basa su estrategia de selección en el algoritmo FIFO, se desea cambiar esta dicha estrategia de selección.

Cada vez que el tiempo de procesador expira para el proceso en ejecución se invoca el método planificar del objeto Planificador en el se encuentra la lógica para seleccionar el siguiente proceso a ejecutar, la actual estrategia es el algoritmo de primero en entrar primero en salir.

- Planificación por turno
- Manejo de listas
- Sobre-escritura de funciones

EL ENTORNO DE DESARROLLO

El entorno de desarrollo de la práctica será el linux.

DESCRIPCIÓN DE LA FUNCIONALIDAD PEDIDA

Se desea que el estudiante sobrescriba la función *planificar* del módulo de planificador de seleccione el siguiente proceso a ejecutar según un turno.

Ejemplo: No Aplica.

CARACTERÍSTICAS DE LA FUNCIÓN

Valor de retorno: void.

Nombre: planificar

Parámetros: void

Acceso: public

```
void planificar();
```

OTROS REQUERIMIENTOS

No Aplica.

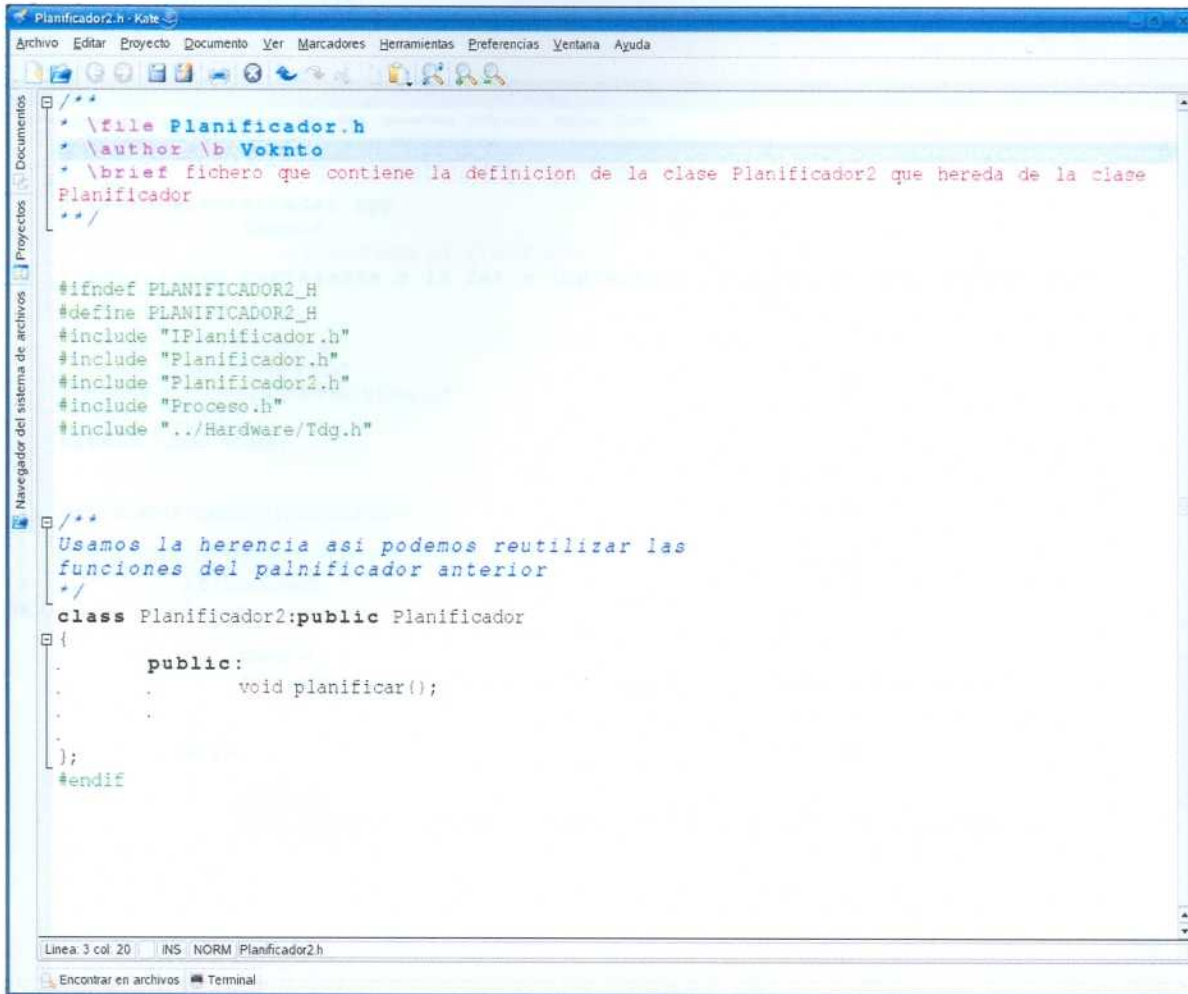
CONCEPTOS UTILIZADOS EN LA PRÁCTICA

- Controlador
- Planificación por turno
- Manejo de listas
- Sobre-escritura de funciones

EL ENTORNO DE DESARROLLO

El entorno de desarrollo de la práctica será Linux.

SOLUCIÓN PROPUESTA



```

/**
 * \file Planificador.h
 * \author \b Voknto
 * \brief fichero que contiene la definicion de la clase Planificador2 que hereda de la clase
        Planificador
 */

#ifndef PLANIFICADOR2_H
#define PLANIFICADOR2_H
#include "IPlanificador.h"
#include "Planificador.h"
#include "Planificador2.h"
#include "Proceso.h"
#include "../Hardware/Tdg.h"

/**
Usamos la herencia asi podemos reutilizar las
funciones del palnificador anterior
*/
class Planificador2:public Planificador
{
    public:
        void planificar();
};

#endif

```

Linea: 3 col: 20 INS NORM Planificador2.h

Encontrar en archivos Terminal

Figura 24 - Planificador2.h

Fuente: Elaboración Propia

3. Implementar el método planificar en la clase Planificador2 modificando el archivo Planificador2.cpp

```

/*
 * \file Planificador.cpp
 * \author \b Voknto
 * \brief fichero que contiene el Planificador
 * Esta clase representa a la fat e implementa la interfaz ISistemadeArchivo
 */

#include "Planificador.h"
#include "Planificador2.h"
#include "../Controladores/Video.h"

extern Video video;

void Planificador2::planificar()
{
    if(turno==0)
    {
        turno=1;
        tdg->setTss((unsigned long)controladorProcesos->getProceso(1).getTss());
    }
    else
    {
        turno=0;
        tdg->setTss((unsigned long)controladorProcesos->getProceso(2).getTss());
    }
}
    
```

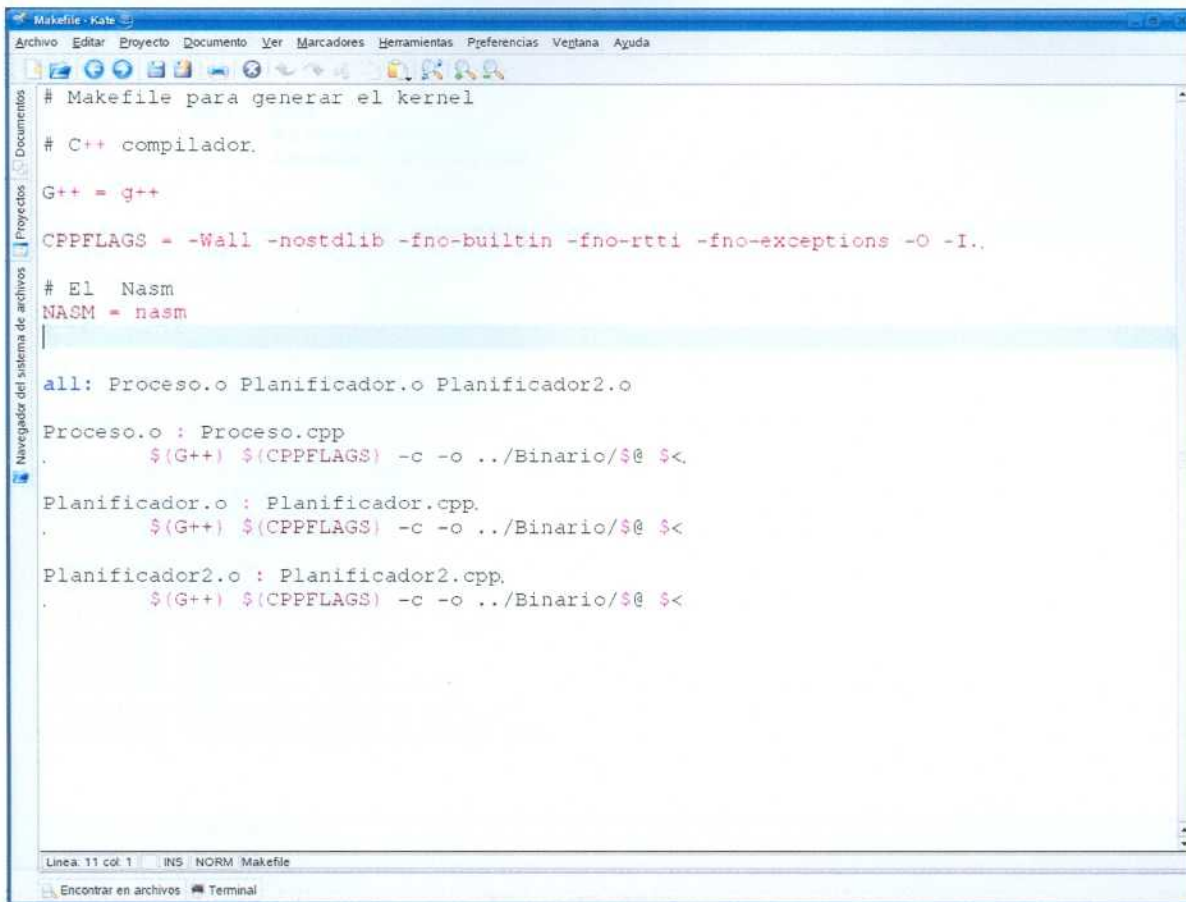
Linea: 17 col: 11 INS NORM Planificador2.cpp

Encontrar en archivos Terminal

Figura 25 - Planificador2.cpp

Fuente: Elaboración Propia

4. Agregar la entrada en el makefile para incluir el planificador2.o



```
# Makefile para generar el kernel

# C++ compilador.
G++ = g++

CPPFLAGS = -Wall -nostdlib -fno-builtin -fno-rtti -fno-exceptions -O -I..

# El Nasm
NASM = nasm

all: Proceso.o Planificador.o Planificador2.o

Proceso.o : Proceso.cpp
. $(G++) $(CPPFLAGS) -c -o ../Binario/$@ $<

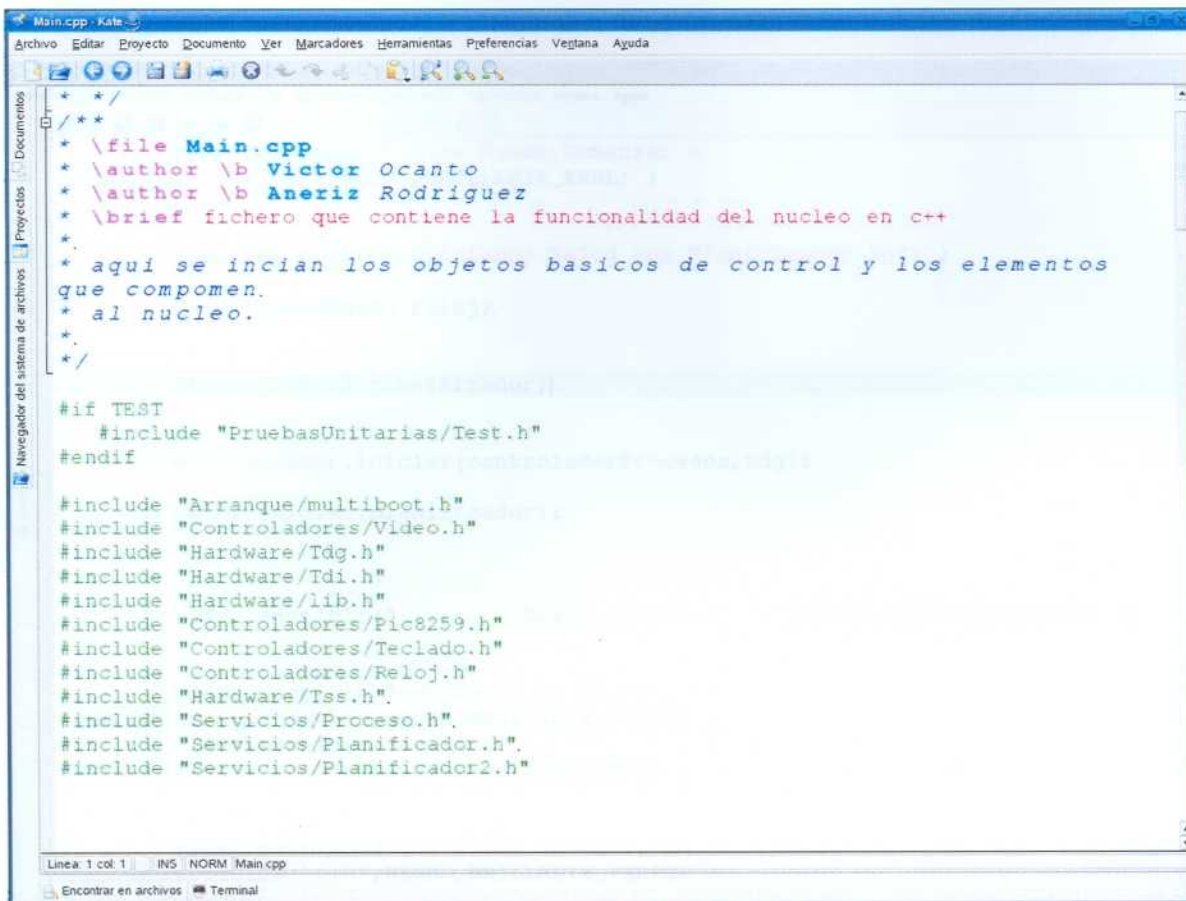
Planificador.o : Planificador.cpp
. $(G++) $(CPPFLAGS) -c -o ../Binario/$@ $<

Planificador2.o : Planificador2.cpp
. $(G++) $(CPPFLAGS) -c -o ../Binario/$@ $<
```

Figura 26 – Makefile

Fuente: Elaboración Propia

5. Agregar en el Main.cpp el uso la inclusión de planificador2.h



```

Main.cpp - Kate
Archivo  Editar  Proyecto  Documento  Ver  Marcadores  Herramientas  Preferencias  Ventana  Ayuda

/**
 * \file Main.cpp
 * \author \b Victor Ocanto
 * \author \b Aneriz Rodriguez
 * \brief fichero que contiene la funcionalidad del nucleo en c++
 *
 * aqui se incian los objetos basicos de control y los elementos
 * que compomen.
 * al nucleo.
 */

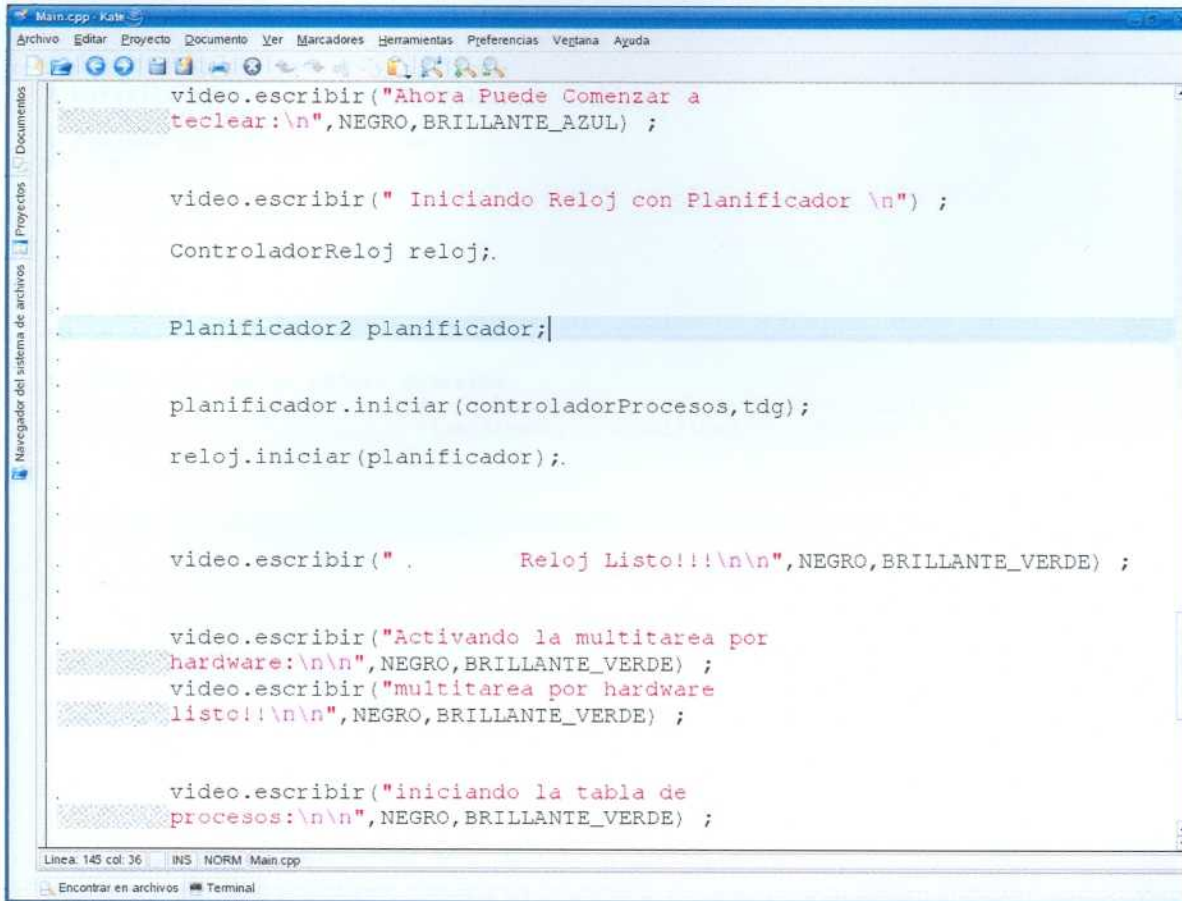
#ifdef TEST
#include "PruebasUnitarias/Test.h"
#endif

#include "Arranque/multiboot.h"
#include "Controladores/Video.h"
#include "Hardware/Tdg.h"
#include "Hardware/Tdi.h"
#include "Hardware/lib.h"
#include "Controladores/Pic8259.h"
#include "Controladores/Teclado.h"
#include "Controladores/Reloj.h"
#include "Hardware/Tss.h"
#include "Servicios/Proceso.h"
#include "Servicios/Planificador.h"
#include "Servicios/Planificador2.h"
    
```

Figura 27 - Main.cpp

Fuente: Elaboración Propia

6. Cambiar la implementación del Planificador a --> Planificador2



```

video.escribir("Ahora Puede Comenzar a
teclear:\n", NEGRO, BRILLANTE_AZUL) ;

video.escribir(" Iniciando Reloj con Planificador \n") ;
ControladorReloj reloj;

Planificador2 planificador;

planificador.iniciar(controladorProcesos, tdg);
reloj.iniciar(planificador);

video.escribir("      Reloj Listo!!!\n\n", NEGRO, BRILLANTE_VERDE) ;

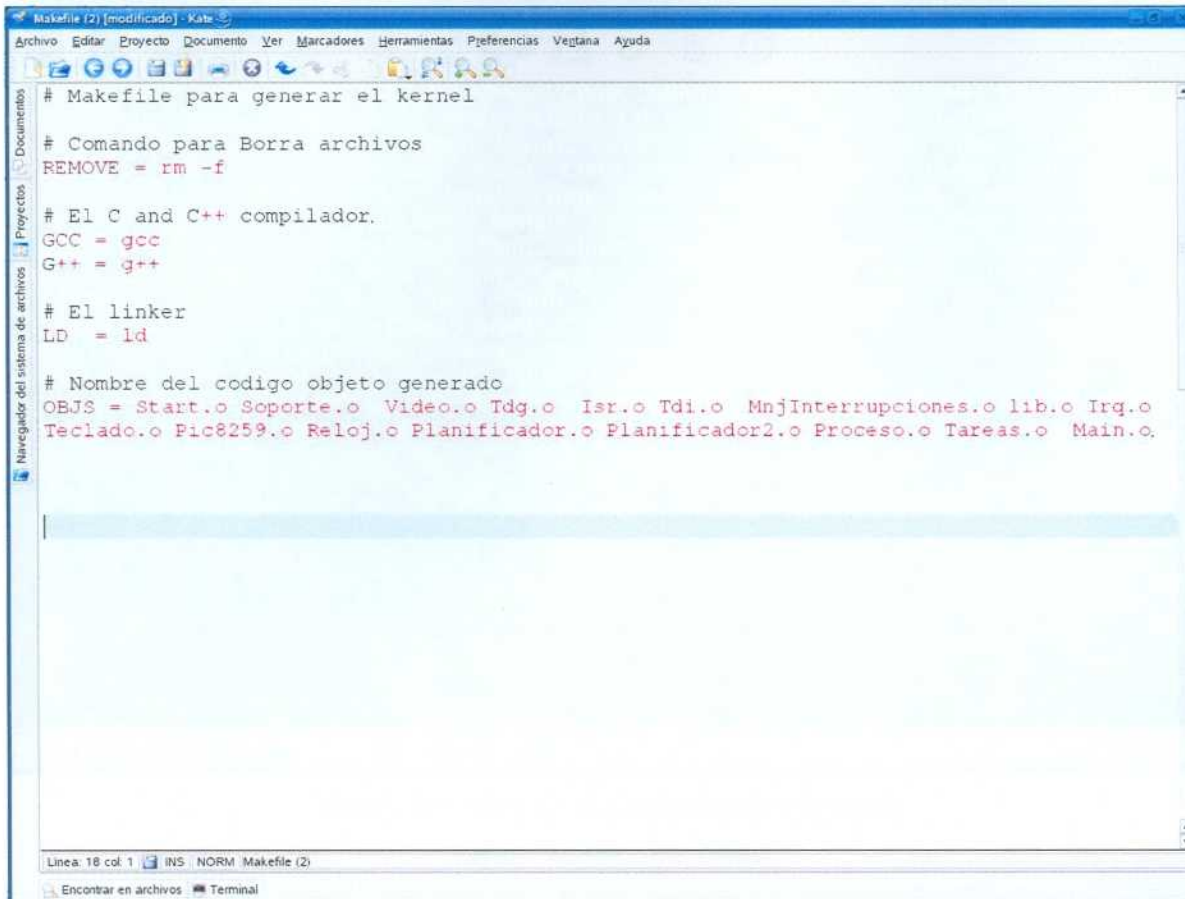
video.escribir("Activando la multitarea por
hardware:\n\n", NEGRO, BRILLANTE_VERDE) ;
video.escribir("multitarea por hardware
listo!!\n\n", NEGRO, BRILLANTE_VERDE) ;

video.escribir("iniciando la tabla de
procesos:\n\n", NEGRO, BRILLANTE_VERDE) ;
    
```

Figura 28 - Main.cpp

Fuente: Elaboración Propia

7. Agregar la entrada en el makefile del kernel para que incluya a Planificador2.o



```

# Makefile para generar el kernel

# Comando para Borra archivos
REMOVE = rm -f

# El C and C++ compilador.
GCC = gcc
G++ = g++

# El linker
LD = ld

# Nombre del codigo objeto generado
OBJS = Start.o Soporte.o Video.o Tdg.o Isr.o Tdi.o MnjInterrupciones.o lib.o Irq.o
Teclado.o Pic8259.o Reloj.o Planificador.o Planificador2.o Proceso.o Tareas.o Main.o
  
```

Figura 29 – Makefile

Fuente: Elaboración Propia

Ejecución del código:

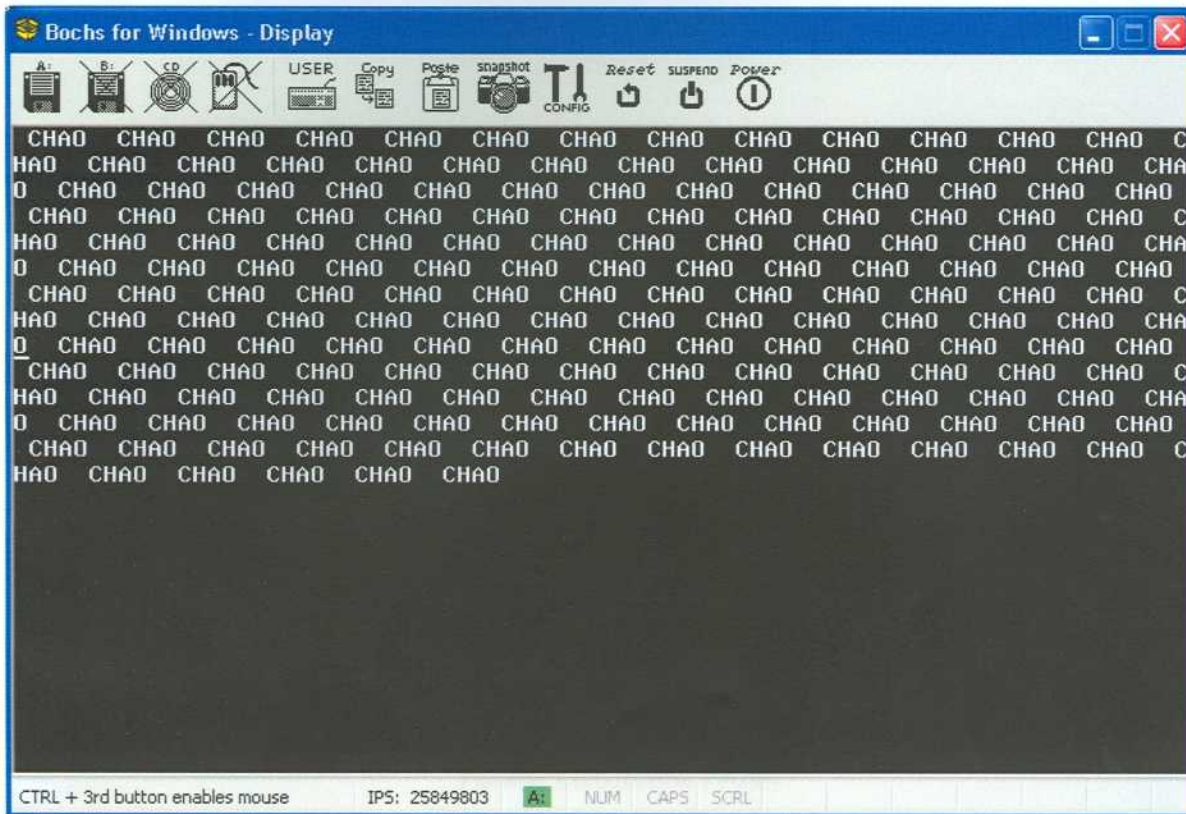


Figura 30 - Ejecución del código (primer proceso)

Fuente: Elaboración Propia

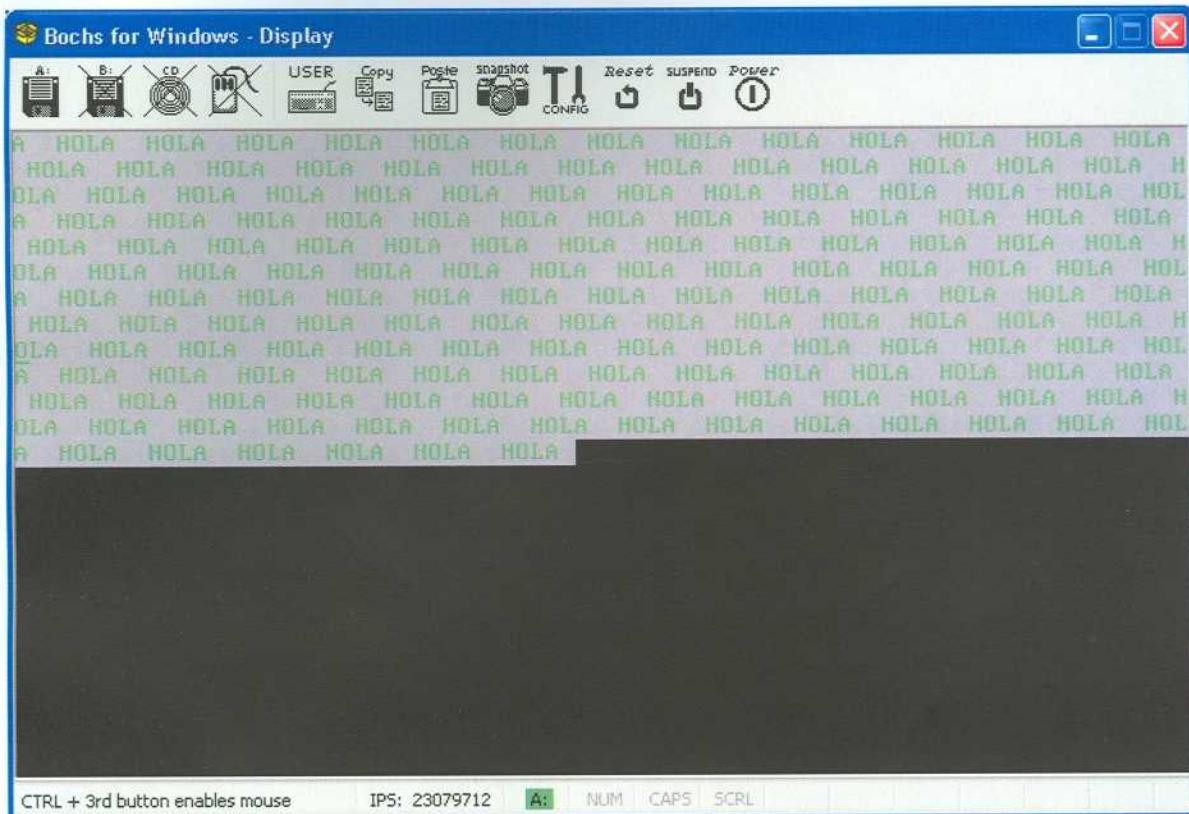


Figura 31 - Ejecución del código (segundo proceso)

Fuente: Elaboración Propia