



UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

PROYECTO DOCENTE CURSO: 2005/06

12710 - DISEÑO DE SISTEMAS OPERATIVOS

ASIGNATURA: 12710 - DISEÑO DE SISTEMAS OPERATIVOS

CENTRO: Escuela de Ingeniería Informática

TITULACIÓN: Ingeniero en Informática

DEPARTAMENTO: INFORMÁTICA Y SISTEMAS

ÁREA: Ciencia De La Comp. E Intel. Artificial

PLAN: 10 - Año 199**ESPECIALIDAD:**

CURSO: Cuarto curso **IMPARTIDA:** Segundo semestre **TIPO:** Obligatoria

CRÉDITOS: 6 **TEÓRICOS:** 3 **PRÁCTICOS:** 3

Descriptores B.O.E.

Análisis y Diseño, Manejadores de Dispositivos. Implementación de Utilidades de Sistema.

Temario

TEMA 1: INTRODUCCIÓN AL LINUX

- 1.1 HISTORIA DE LINUX
 - 1.2 LICENCIA GNU UNA NUEVA FILOSOFÍA
 - 1.3 ARQUITECTURAS DE UN NÚCLEO
 - 1.4 ESTRUCTURA DEL NÚCLEO
 - 1.5 SÍNTESIS Y UTILIZACIÓN DE LLAMADAS
- BIBLIOGRAFÍA: MAX-99, CAR-97
TEMPORIZACIÓN: 2h

TEMA 2: INTERRUPTIONES

- 2.1 INTRODUCCIÓN
 - 2.2 INTERRUPTIONES HARDWARE
 - 2.3 INTERRUPTIONES SOFTWARE
 - 2.4 INTERRUPTIONES POR SEÑALES
- BIBLIOGRAFÍA: MAX-99, CAR-97
TEMPORIZACIÓN: 2h

TEMA 3: CARGA DEL NÚCLEO E INICIALIZACIÓN

- 3.1 INTRODUCCIÓN
 - 3.2 CARGA DEL NÚCLEO LILO
 - 3.3 EL PROGRAMA DE INICIALIZACIÓN INIT
 - 3.4 EL PROGRAMA PRINCIPAL DEL NÚCLEO MAIN
- BIBLIOGRAFÍA: MAX-99, CAR-97
TEMPORIZACIÓN: 1h

TEMA 4: LLAMADAS AL SISTEMA

- 4.1 INTRODUCCIÓN
- 4.2 MANEJO DE PROCESOS
- 4.3 SEÑALES

4.4 MANEJO DE FICHEROS

4.5 MANEJO DE DIRECTORIOS Y SISTEMA DE FICHEROS

4.6 PROTECCIÓN

BIBLIOGRAFÍA: BAC-86, MAR-96, TAN-98, ROC-85

TEMPORIZACIÓN: 1h

TEMA 5: PROCESOS

5.1 CONCEPTOS SOBRE PROCESOS

5.2 TABLA DE PROCESOS

5.3 GESTIÓN Y PLANIFICACIÓN DE PROCESOS

5.4 CREACIÓN DE PROCESOS FORK, CLONE

5.5 EJECUCIÓN DE UN PROCESO EXEC

5.6 TERMINACIÓN DE UN PROCESO EXIT

5.7 ESPERA POR UN PROCESO WAIT

BIBLIOGRAFÍA: MAX-99, CAR-97, TAN-98

TEMPORIZACIÓN: 4h

TEMA 6: MEMORIA

6.1 PANORAMA DEL MANEJO DE MEMORIA EN LINUX

6.2 MEMORIA VIRTUAL DE UN PROCESO

6.3 PAGINACIÓN

6.4 INTERCAMBIO DE PAGINAS CON EL DISCO SWAP

6.5 VOLCADO DE UN FICHERO EN MEMORIA MMAP

6.6 ASIGNACIÓN DE ESPACIO DE DATOS BRK

BIBLIOGRAFÍA: MAX-99, CAR-97

TEMPORIZACIÓN: 5h

TEMA 7: ARCHIVOS

7.1 SISTEMA VIRTUAL DE ARCHIVOS

7.2 BUFFER CACHE.

7.3 TABLA DE INODOS.

7.4 TABLA DE SUPERBLOQUE.

7.5 DESCRIPTOR DE UN FICHERO.

7.6 APERTURA DE UN FICHERO OPEN, CLOSE

7.7 OPERACIONES SOBRE FICHEROS READY WRITE

7.8 MONTAR UN SISTEMA DE FICHEROS MOUNT.

BIBLIOGRAFÍA: MAX-99, CAR-97

TEMPORIZACIÓN: 5h

TEMA 8: COMUNICACIÓN ENTRE PROCESOS

8.1 PRIMITIVAS DE COMUNICACIÓN

8.2 MENSAJES

8.3 SEMAFOROS

8.4 MEMORIA COMPARTIDA

8.5 PIPE

8.6 SOCKETS

BIBLIOGRAFÍA: MAX-99, CAR-97

TEMPORIZACIÓN: 5h

TEMA 9: MANEJADORES DE DISPOSITIVOS. DRIVERS

9.1 ENTRADA SALIDA

9.2 CONSOLA

9.3 TECLADO
9.4 IMPRESORA
9.5 RAMDISK
9.6 DISCOS IDE
9.7 CDROM
TEMPORIZACIÓN: 5h

Conocimientos Previos a Valorar

- Sistemas Operativos (SO)
- Organización de computadores (FC, SD)
- Programación estructurada y modular
- Tipos abstractos de datos (ED1, MP1)
- Lenguaje C, a nivel principiante-intermedio (ED2)

Objetivos

Los principales objetivos didácticos que se tratan en este curso de Diseño de Sistemas Operativos, tratan con: Afianzar los conceptos básicos estudiados en asignaturas anteriores de sistemas operativos. Estudio de las estructuras de datos internas de un sistema operativo real, estudio de las funciones y procedimientos internos del núcleo de un sistema operativo real, análisis de manejadores de dispositivos. Planteamiento y resolución de problemas prácticos derivados de una implementación real, utilizando las llamadas y librerías del sistema. Diseño e implementación de utilidades del sistema. Estudio de temas relacionados con los últimos avances en sistemas operativos. Presentamos un desglose de objetivos por temas según programa:

TEMA 1: INTRODUCCIÓN AL LINUX

El primer capítulo, realiza una descripción de la arquitectura y estructura del núcleo. También se describe los pasos que se siguen en una llamada al sistema.

TEMA 2: INTERRUPCIONES

Los objetivos de este capítulo tratan de afianzar el concepto de interrupción y los pasos que ejecuta el núcleo cuando se realiza una interrupción. Se estudia el mecanismo de interrupción software de las llamadas al sistema, sus pasos, estructuras y procedimientos, así como las excepciones generadas en la CPU. Se estudia como implementa Linux el sistema de interrupciones hardware, se analizan las estructuras de datos y procedimientos que permiten gestionar y atender las interrupciones hardware. Finalmente se estudia el sistema de señales, las estructuras y los procedimientos que gestionan el envío y recepción de señales.

TEMA 3: ARRANQUE E INICIALIZACIÓN DEL NÚCLEO

Los objetivos del segundo capítulo tratan de la comprensión de todos los pasos que ocurren desde que se enciende el computador, hasta que este presenta el símbolo del procesador de comandos. Analizar y estudiar el programa de inicialización INIT.

TEMA 4: LLAMADAS AL SISTEMA

El tercer capítulo está relacionado con las llamadas al sistema, que están soportadas por LINUX, haciéndose un estudio funcional de cada una de ellas y agrupándolas según su aplicación a procesos, señales, manejo de ficheros, relativas a directorios, protección y manejo de tiempos.

TEMA 5: PROCESOS

El siguiente conjunto de capítulos trata sobre los bloques fundamentales que conforman un S.O. comenzando con el bloque que estudia los procesos. En él se hace un repaso a los conceptos sobre procesos, su estructura interna, pasándose a continuación a estudiar su realización. Se analizan las

estructuras de datos para el manejo de procesos. Se estudia la implementación de estructuras y procedimientos para la gestión de CPU. Se estudian llamadas al sistema que están relacionadas con la creación de procesos FORK y CLONE, con la ejecución de la imagen de un programa EXEC, la terminación de un proceso EXIT y la espera por un proceso WAIT.

TEMA 6: MEMORIA

En el siguiente capítulo sobre memoria, se trata de estudiar las estructuras y procedimientos que asignan memoria virtual a un proceso, se analiza la técnica de paginación las estructuras de tablas de páginas y los procedimientos que las manejan, se estudian los procedimientos que permiten tratar el disco como una prolongación de la memoria SWAP. Se analizan las llamadas que permiten proyectar un fichero de disco en memoria, MMAP. Se termina estudiando llamadas relacionadas con el manejo de memoria como BRK.

TEMA 7: ARCHIVOS

Este capítulo versa sobre ficheros donde se estudia el sistema de ficheros virtual implementado por Linux, con sus estructuras de datos principales, el bloque cache, el superbloque, el inodo, y la estructura de directorio. Se estudia uno de los sistemas de ficheros reales que soporta Linux, la segunda parte de este capítulo está relacionada con la implementación de las operaciones sobre ficheros, CREAT, MAKENOD, OPEN, CLOSE READ, WRITE y SEEK, el manejo de directorios y la implementación de las llamadas, MOUNT, UMount, así como otras llamadas como CHDIR, CHROOT, STAT y FSTAT.

TEMA 8: COMUNICACIÓN ENTRE PROCESOS

Los objetivos de este capítulo tratan de estudiar las primitivas de comunicación entre procesos PIC. Se estudian los mecanismos de mensajes, pipe, semaforos y memoria compartida, con las estructuras de datos y procedimientos que los implementan.

TEMA 9: MANEJADORES DE DISPOSITIVOS. DRIVERS

En este capítulo se estudian varios manejadores de dispositivos relacionados con el modo carácter como el terminal, el teclado, la consola y varios manejadores de modo bloque como discos y cdrom.

Metodología de la Asignatura

En las clases teóricas se explicarán los fundamentos sobre diseño de sistemas operativos descritos en el temario. La teoría se impartirá mediante clases magistrales basadas en los textos de referencia de la asignatura. El estudiante dispondrá de apuntes de cada unidad temática para que conozca los puntos más importantes de ella.

Prácticas

Durante el curso se realizan seis prácticas, en el laboratorio asignado a la asignatura de Diseño de Sistemas Operativos, situado en el Módulo 2, del edificio de Informática, sobre Linux. La realización de estas prácticas persigue varios objetivos, primero introducir al alumno en las herramientas de instalar, compilar y generar un nuevo sistema operativo, segundo el diseño e implementación de aplicaciones de administrador de sistemas, utilizando las llamadas al sistema y utilidades que proporciona el sistema operativo y tercero generar nuevos manejadores y llamadas al sistema. El profesor suministra enunciados, información, y metodología para realizar las prácticas. Estas prácticas realizadas en grupos de dos, son obligatorias y necesarias para poder presentarse a los exámenes.

Trabajos

Los alumnos en grupos de dos, o individualmente, elaboran trabajos sobre la asignatura, uno referente a la primera parte de la asignatura donde se estudian las primeras capas del sistema

operativo y otro referente a la segunda parte de la asignatura. Estos trabajos se exponen y discuten en clase y son obligatorios para poder presentarse a los exámenes. El profesor suministra el asesoramiento, enunciados, material bibliográfico, y metodología para realizar los trabajos y para su exposición en clase.

Se promoverán aspectos colaterales de formación, como la capacidad expresiva (oral y escrita) y la calidad de programación. Se promueve el uso de tecnologías avanzadas (web, correo electrónico) como instrumento de comunicación estudiante/profesor y como depósito de información (p.ej. las guías, manuales y fichas estarán disponibles en formato electrónico).

Planificación Temporal

SEMANA, TEORÍA, PRÁCTICA

- 1 T1 P1
- 2 T2 P2
- 3 T3,T4 P2
- 4 T5 P3
- 5 T5 P3
- 6 T6 P3
- 7 T6 P3
- 8 T6,T7 P3
- 9 T7 P4
- 10 T7 P5
- 11 T8 P5
- 12 T8 P6
- 13 T8,T9 P6
- 14 T9 P7
- 15 T9 P7

Evaluación

Evaluación de los conocimientos teóricos.

Para evaluar los conocimientos teóricos de la asignatura, además de las convocatorias finales oficiales que fija la Universidad, ordinario, extraordinario (septiembre) y extraordinario especial (diciembre), se realizan dos exámenes parciales durante el curso, uno correspondiente a la primera parte de la asignatura y otro examen parcial de la segunda parte de la asignatura. La superación de un examen parcial, libera al alumno de esa parte de la asignatura para el examen ordinario, (si bien puede presentarse al examen final para mejorar nota). Los exámenes parciales no liberan materia en los exámenes extraordinarios. Es condición necesaria aprobar la teoría para poder aprobar la asignatura.

Evaluación de los trabajos.

El alumno debe realizar trabajos correspondientes a las materias del curso, estos trabajos son evaluados cuando se exponen y discuten en clase junto con la entrega de la memoria escrita del mismo. La realización de los trabajos es obligatoria y la aprobación de los mismos es condición necesaria para poder aprobar la asignatura. Una vez superados los trabajos, su calificación se guarda durante el curso en que se realizaron, esto es Julio, Septiembre y Diciembre, no para siempre.

Evaluación de las prácticas.

Para evaluar los conocimientos prácticos, se prueban y testean en el laboratorio cada una de las prácticas y se defiende la memoria de las mismas. Estas prácticas son obligatorias y necesarias para poder aprobar la asignatura. Una vez superadas las prácticas, su calificación se guarda mientras no cambie el proyecto docente.

Evaluación de otras actividades.

Durante el curso se tendrá en cuenta otros factores como asistencia, realización de test o ejercicios, participación en la asignatura, actitud positiva en clase, ayuda a los compañeros, etc.

Evaluación en las convocatorias extraordinarias.

En las convocatorias extraordinarias se realizará un examen escrito y una entrega y defensa de los trabajos para evaluar los conocimientos teóricos; y una entrega de prácticas y defensa de las mismas en el laboratorio para evaluar los conocimientos prácticos.

Calificación final.

Para la calificación final se tienen en cuenta las practicas, los trabajos expuestos en clase, los exámenes parciales y otros, de modo que el peso de cada uno de estos aspectos en la nota final, expresado en porcentajes es el siguiente:

Prácticas 20 %. Trabajos 20 %. 1er Parcial 30 %. 2º Parcial 30 %. Otros 10 % (Mejorar Nota).

Descripción de las Prácticas

PRÁCTICAS

Durante el curso se realizan seis prácticas, en el laboratorio asignado a la asignatura de Diseño de Sistemas Operativos, situado en el Modulo 2-2, del edificio de Informática, sobre Linux. La realización de estas prácticas persigue varios objetivos, primero introducir al alumno en las herramientas de compilar y generar un nuevo sistema operativo, segundo el diseño e implementación de aplicaciones de administrador de sistemas, utilizando las llamadas al sistema y utilidades que proporciona el sistema operativo y tercero generar nuevos manejadores y llamadas al sistema. El profesor suministra enunciados, información, y metodología para realizar las prácticas. Estas prácticas realizadas en grupos de dos, son obligatorias y necesarias para poder presentarse a los exámenes.

PRÁCTICA 1: INSTALACIÓN DE LINUX

DESCRIPCIÓN

- Instalar Linux en disco duro, haciendo varias particiones.
- Arrancar el Linux y realizar un recorrido por sus directorios visualizando los ficheros que hay en cada uno de ellos.
- Crear un nuevo usuario con un nombre personal y contraseña el DNI.
- Utilizar los comandos del Linux y estudiar las funciones que realizan.

OBJETIVOS

- Realizar una Instalación típica de Linux
- Manejar los parámetros de una instalación
- Familiarizarse con la utilización del Linux
- Recorrido por la estructura de sistema de ficheros
- Aprendizaje de los comandos

Nº HORAS estimadas en laboratorio: 2

PRÁCTICA 2: ALARMA

DESCRIPCIÓN

- Construya un programa principal, que utilice el recurso pantalla continuamente (realice la función que usted quiera, por ejemplo que imprima su nombre iterativa mente, que dibuje una pelota

moviéndose por la pantalla, etc.), pero que cada 4 segundos de forma repetitiva lance otro programa que deberá mostrar la fecha y hora actual (ejecutar un date), en el recurso pantalla.

Ayúdese de las llamadas FORK EXEC, WAIT, SIGNAL, y ALARM.

- Plantee una solución basada en Sección Crítica, para que no exista conflicto en la utilización del recurso pantalla por parte de los programas, el principal y el que lanza la hora.
- Explique en una memoria el programa Alarma, con el listado en C, debidamente documentado.

OBJETIVOS

- Utilización del tiempo real y reloj del sistema.
- Manejo de las llamadas al sistema que trabajan con creación de procesos, ejecución de la imagen de un proceso, espera por un proceso y envío y recepción de señales.
- Estudio del problema de la Sección Crítica.

Nº HORAS estimadas en laboratorio: 4

PRÁCTICA 3: DISEÑO E IMPLEMENTACIÓN DE UN PROCESADOR DE COMANDOS

DESCRIPCIÓN

•Realizar un programa utilizando las llamadas al sistema de manejo de procesos fork, wait, exec, que permita ejecutar cualquier comando del sistema operativo, o lanzar otro programa, que admita pipe, redireccionar, lanzar en modo tanda. Sacrifique ahorro de código por modularidad y claridad.

- Explique en una memoria el programa Shell, con las siguientes líneas maestras:
- Diagrama de diseño y funcionamiento con todos los bloques que lo componen.
- Descripción de cada uno de los bloques.

Para cada bloque, función que realiza, parámetros de entrada y salida, estructuras de datos que emplea, funciones que utiliza.

- Listado en C del programa debidamente documentado.

OBJETIVOS

- Utilización de las llamadas al sistema, FORK, WAIT, PIPE, EXEC.
- Diseño de la estructura de un Shell

Nº HORAS estimadas en laboratorio: 10

PRÁCTICA 4: MAPEAR UN FICHERO EN MEMORIA MMAP

DESCRIPCIÓN

- Crear un fichero de texto
- Abrir el fichero
- Proyectar el fichero en memoria mmap
- Modificar el contenido del fichero, pudiendo cambiar su tamaño
- Liberar munmap y cerrar el fichero
- Visualizar los cambios del fichero
- Presentar una memoria explicando el programa y comentando el código en C.

OBJETIVOS

- Profundizar y afianzar los conceptos de proyectar un fichero en memoria.
- Aprender a utilizar la llamada que hace posible el mapeo de un fichero en memoria mmap.
- Aprender la llamada al sistema que deshace el mapeo de un fichero en memoria munmap.

Nº HORAS estimadas en laboratorio: 2

PRÁCTICA 5: COMPILAR EL NÚCLEO

DESCRIPCIÓN

•Compilar los fuentes de Linux, que ya existen para crear un nuevo núcleo, aprender los distintos parámetros que definen una compilación y el comando MAKE.

OBJETIVOS

- Aprender a compilar el núcleo
- Utilizar el comando Make
- Comprender los tipos de compilación
- Generar un nuevo Linux.

Nº HORAS estimadas en laboratorio: 4

PRÁCTICA 6: CREAR UNA NUEVA LLAMADA AL SISTEMA

DESCRIPCIÓN

- Diseñar una nueva llamada al sistema que al menos maneje tres parámetros de entrada y uno de salida, utilizando las facilidades del sistema como _syscall3
- Compile el núcleo con la nueva llamada y realice un programa para utilizarla.
- Explique en una memoria todos los pasos realizados, con el listado en C, debidamente documentado.

OBJETIVOS

- Diseñar una nueva llamada al sistema
- Implementar dentro del núcleo la llamada

Nº HORAS estimadas en laboratorio: 4

PRÁCTICA 7: MEJORAR EL MANEJADOR DE MEMORIA

DESCRIPCIÓN

- Construir un nuevo servidor de memoria.
- Modificar el servidor de memoria con nuevas características y construir un nuevo disco boot con este servidor.
- Mejorar el sistema de asignación de memoria a un proceso, cambiando el algoritmo que actualmente tiene implementado de First Fit, el primero que cabe. Por ejemplo, implementar el algoritmo Best Fit el mejor que cabe, para la asignación de espacio de memoria.
- Entregar un listado indicando todas las partes de Linux modificadas.
- Construir un nuevo disco boot con el nuevo servidor de memoria.

OBJETIVOS

- Mejorar el servidor de memoria con nuevos algoritmos.
- Construir un nuevo servidor de memoria.
- Compilar el núcleo con el manejador de memoria.
- Generar un nuevo disco boot con el sistema operativo.

Nº HORAS estimadas en laboratorio: 4

MATERIAL DE LABORATORIO RECOMENDADO (SOFTWARE) para todas las prácticas.

. Sistema Operativo Linux, paquete ofimática, salida a Internet.

MATERIAL DE LABORATORIO RECOMENDADO (HARDWARE) para todas las prácticas.
. Puestos de trabajo basados en PC's, Pentium, EXCLUSIVOS, para que los alumnos puedan instalar y compilar con privilegios de root.

Bibliografía

[1] Operating system concepts /

Abraham Silverschatz, James L. Peterson.
Addison-Wesley,, Reading (Massachusetts) : (1988)
0201187604

[2] Linux Device Drivers

Alessandro Rubini
O'Reilly Associates INC - (1996)

[3] Sistemas operativos modernos /

Andrew S. Tanenbaum.
Prentice Hall,, México : (2003) - (2ª ed.)

[4] Sistemas operativos: diseño e implementación /

Andrew S. Tanenbaum.
Prentice-Hall,, México : (1998) - (2ª ed.)
*970-17-0165-8**

[5] The Design of the UNIX Operating System

Basch
Prentice Hall - (1986)

[6] IA-64 Linux Kernel: design and implementation /

David Mosberger, Stéphane Eranian.
Prentice Hall,, Upper Saddle River (New Jersey) : (2002)
0-13-061014-3

[7] UNIX: programación avanzada /

Francisco Manuel Márquez.
Ra-Ma,, Madrid : (1996) - (2ª ed.)
8478972390

[8] Linux kernel internals /

Michael Beck...[et al.].
Addison-Wesley,, Harlow : (1998) - (2nd ed.)
0201331438

[9] PC interno 5 : programación de sistemas /

Michael Tischer, Bruno Jennrich.
Marcombo,, Barcelona : (1996)
8426710816

[10] The Linux Kernel book /

Rémy Card ; Eric Dumas ; Franck Mével.
John Wiley & Sons,, Chichester : (2000)
0471981419

[11] Anatomia del Núcleo de Linux 2.4

S.Candela, R.García, G.Padrón

Dto Publicaciones de la ULPGC - (2003)

en tramite 84-96131-57-2

[12] Linux Core Kernel commentary /

Scott Maxwell.

The Coriolis Group,, Scottsdale, Arizona : (1999)

1576104699

Equipo Docente

SANTIAGO CANDELA SOLÁ

(COORDINADOR)

Categoría: *CATEDRATICO DE UNIVERSIDAD*

Departamento: *INFORMÁTICA Y SISTEMAS*

Teléfono: *928458753* **Correo Electrónico:** *santiago.candela@ulpgc.es*