



UNIVERSIDAD DE LAS PALMAS  
DE GRAN CANARIA

GUÍA DOCENTE

CURSO: 2012/13

## 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

### Información ECTS

Créditos ECTS:6

Horas de trabajo del alumno:(140-150)

Horas presenciales:60

- Horas teóricas (HT):30

- Horas prácticas (HP):30

- otras:

Horas no presenciales:80

- trabajos tutorizados (HTT):20

- actividad independiente (HAI):60

Idioma en que se imparte: español

### Descriptores B.O.E.

Análisis y Diseño de 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 SINTESIS Y UTILIZACIÓN DE LLAMADAS

##### BIBLIOGRAFÍA:

Understanding the Linux Kernel

The Linux Kernel book

Linux Core Kernel commentary

TEMPORIZACIÓN: 2h

#### TEMA 2: INTERRUPCIONES

##### 2.1 INTRODUCCIÓN

##### 2.2 INTERRUPCIONES HARDWARE

##### 2.3 INTERRUPCIONES SOFTWARE

## 2.4 INTERRUPTACIONES POR SEÑALES

### BIBLIOGRAFÍA:

Understanding the Linux Kernel

The Linux Kernel book

Linux Core Kernel commentary

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:

Understanding the Linux Kernel

The Linux Kernel book

Linux Core Kernel commentary

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:

UNIX programación avanzada

The UNIX programming environment

Linux kernel programming

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:

Understanding the Linux Kernel

The Linux Kernel book

Linux Core Kernel commentary

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:

Understanding the Linux Kernel  
The Linux Kernel book  
Linux Core Kernel commentary  
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:

Understanding the Linux Kernel  
The Linux Kernel book  
Linux Core Kernel commentary  
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:

Understanding the Linux Kernel  
The Linux Kernel book  
Linux Core Kernel commentary  
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

#### BIBLIOGRAFÍA:

Understanding the Linux Kernel  
The Linux Kernel book  
Linux device drivers  
TEMPORIZACIÓN: 5h

## Requisitos Previos

- Sistemas Operativos (SO)
- Organización de computadores (FC, SD)
- Programación estructurada y modular
- Tipos abstractos de datos (ED1, MP1)

## Objetivos

Objetivos y competencias de conocimientos, habilidades y destrezas.

- Adquisición de Conocimientos. Afianzar los conceptos básicos estudiados en asignaturas anteriores de sistemas operativos. Estudio de las estructuras internas de un sistema operativo real, estudio de las funciones y procedimientos internos del núcleo de un sistema operativo real. Estudio de temas relacionados con los últimos avances en sistemas operativos.
- Adquisición de Destrezas y Habilidades. Planteamiento (análisis) y resolución (síntesis) de problemas prácticos derivados de una implementación real, utilizando las llamadas y librerías del sistema. Diseño e implementación de utilidades del sistema.

Objetivos y competencias de herramientas de aprendizaje, formación en equipo, y capacidades individuales.

- Aprender a Aprender. Se plantean tareas para que los estudiantes aprendan a buscar información utilizando bibliografía y las nuevas tecnologías de la información. Aprendan a analizar la información, filtrarla, sintetizarla, contrastarla y aprenderla.
- Trabajo en Grupo. Se desarrollan actividades para trabajar en grupo, donde se aprenda a distribuir trabajo, cooperación y puesta en común del mismo, desarrollo de las relaciones interpersonales y adquirir destrezas sociales.
- Potenciación de la expresión oral y escrita. Mediante la exposición de trabajos en clase utilizando herramientas de presentación, se desarrolla la presentación oral en público, desarrollando la autoestima, confianza consigo mismo y pérdida de la timidez en grupo. La entrega de trabajos escritos a la clase potencia la escritura y modo de expresión.
- Utilizar las Nuevas Tecnologías de la Información. Aprender el uso de nuevas plataformas educativas basadas en las tecnologías de la información y la comunicación para su uso el proceso enseñanza-aprendizaje.

El módulo Diseño del Núcleo de un Sistema Operativo está basado en el estudio del Sistema Operativo LINUX desarrollado por Linus Torvalds.

La disponibilidad de los programas fuentes de este S.O. Así como libros y documentación en Internet, proporcionan una buena herramienta pedagógica para cubrir los objetivos de esta materia, permitiendo el análisis, síntesis, desarrollo y prueba de nuevos módulos que conforman parte de un S.O.

Desglose de competencias y objetivos por temas:

### 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, a si 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, OPEN, CLOSE READ, WRITE y SEEK, el manejo de directorios y la implementación de las llamadas, MOUNT, UMount, así como otras llamadas.

#### 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

Para alcanzar los objetivos reseñados, se presenta una metodología basada en varias estrategias con diversas actividades para conseguir la participación e implicación de los estudiantes en la asignatura. Un marco de trabajo donde los estudiantes adquieran un protagonismo activo y adquieran una autonomía en su aprendizaje, y el profesor como experto realice una actividad de dirección, marque líneas de trabajo, tutor de la labor del estudiante, analice su trabajo, lo critique y lo valore.

## EXPOSICIONES

En las clases en el aula se explican los fundamentos sobre diseño de sistemas operativos descritos en el temario. La teoría se imparte mediante clases magistrales basadas en los textos de referencia de la asignatura, bibliografía y búsqueda de información en Internet. El estudiante dispondrá de apuntes de cada unidad temática para que conozca los puntos más importantes de ella. En las clases de teoría en el aula, los estudiantes expondrán y presentarán sus trabajos partiendo de un caso real, para su discusión, análisis y aprendizaje. Se promueven aspectos colaterales de formación, como la capacidad expresiva (oral y escrita) y la calidad de programación.

## TRABAJOS EN GRUPO

Los estudiantes 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.

## PRÁCTICAS

Durante el curso se realizan prácticas sobre Linux, en el laboratorio asignado a la asignatura de Diseño de Sistemas Operativos, situado en el Modulo 2, del edificio de Informática. 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.

## TRABAJO AUTÓNOMO

Se precisa de un trabajo autónomo, donde el estudiante analice, reflexione, comprenda y memorice conocimientos de la materia.

## APOYO A LA ENSEÑANZA PRESENCIAL MEDIANTE LAS TIC

Se promueve el uso de tecnologías de la información y comunicación, ( utilización de la plataforma Moodle, web, Internet) como instrumento de comunicación estudiante-profesor y estudiante-estudiante. Como depósito de información (p.ej. las guías, manuales y fichas están disponibles en formato electrónico a través de la plataforma moodle). Esta plataforma permite romper el espacio, el estudiante puede trabajar en cualquier lugar que tenga un ordenador conectado a internet, y rompe el tiempo en el sentido que el estudiante tiene a su disposición el curso las veinticuatro horas del día. Todo ello favorece el proceso de enseñanza-aprendizaje dotándolo de mayor flexibilidad, facilidad de acceso a la información y el trabajo cooperativo.

## ACTIVIDADES EXTRA ACADÉMICAS

Se promueve durante el curso actividades extra académicas propuestas, organizadas y realizadas por los estudiantes, como actividades de la naturaleza, con el objetivo de crear un buen ambiente en el curso, una mayor amistad entre los estudiantes, y un mejor conocimiento social de los miembros del curso.

El desarrollo del curso se realiza mediante actividades que podemos clasificar de forma presencia y no presencial veamos estas actividades:

## ACTIVIDADES PRESENCIALES

Clases en el aula. Esta actividad se realiza en el aula y es donde se explican los contenidos de la materia cuentan con la presencia del profesor y se desarrollan por el profesor y los estudiantes supervisados por el profesor.

Clases en el laboratorio. Estas clases se realizan en el laboratorio, donde los estudiantes tienen asignado un puesto de trabajo basado en un ordenador conectado en red y el software necesario para desarrollar las prácticas. Cuentan con la presencia del profesor, para explicar las prácticas y resolver las dudas y cuestiones.

Tutorías. Esta actividad se realiza de forma presencial en el despacho del profesor, donde el profesor entrega material, explica las directrices y guía de los trabajos a realizar, resuelve dudas, y se mantiene una relación individual mas personalizada, donde el estudiante puede expresar cuestiones, opiniones o forma de pensar que no es capaz de realizar en el ámbito general de la clase.

Exámenes. Esta actividad se realiza de forma presencial, bien en aula (exámenes teóricos) donde el estudiante muestra los conocimientos adquiridos de forma escrita, o bien en el laboratorio (exámenes de prácticas) donde el estudiante presenta, defiende y prueba las prácticas en el ordenador delante del profesor.

Actividades extra académicas. Si bien estas actividades se realizan fuera del aula, se requiere la presencia tanto de los estudiantes como del profesor para potenciar la relación de los miembros de la asignatura.

## ACTIVIDADES NO PRESENCIALES

Adquisición de los conocimientos teóricos. Trabajo autónomo del estudiante donde estudia, analiza, sintetiza y adquiere los conocimientos de la materia. Elabora presentaciones y memoria para las exposiciones.

Adquisición de conocimientos prácticos. Trabajo en grupo e individual donde el estudiante bien en el laboratorio (están disponibles cuando no hay clases presenciales), o en las salas de informática del edificio, o bien en el ordenador propio de casa, realiza trabajos prácticos, elabora documentación para las memorias de prácticas.

Preparación de exámenes. El estudiante de forma autónoma prepara los exámenes según las directrices dadas por el profesor, realizando sus diagramas, mapas, resúmenes, etc.

Tutorías on-line. Se utiliza el entorno virtual, correo, foro etc, para realizar tutorías no presenciales, de forma que el estudiante pueda realizar preguntas o consultas en cualquier momento.

## 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

## **Criterios de 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 (julio) 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 dos 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 tes 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 tiene en cuenta las prácticas, los trabajos expuestos en clase, y los exámenes parciales y/o finales. Existen otras actividades que denominamos “otros”, como mantener una actitud participativa en clase, buen compañerismo, participar en actividades extra académicas que pueden mejorar la nota, de modo que el peso de cada uno de estos aspectos en la nota final, expresado en porcentajes es el siguiente:

40% Prácticas, 20% Trabajos, 20% 1er Parcial,  
20% 2º Parcial, 10% Otros(Mejorar Nota)



### PRÁCTICAS

Durante el curso se realizan prácticas, en el laboratorio asignado a la asignatura de Diseño de Sistemas Operativos, situado en el Modulo-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 Básico] Linux device drivers /

*Alessandro Rubini & Jonathan Corbet.*  
*O'Reilly and Associates., Beijing : (2001) - (2nd ed.)*  
 0-596-00008-1

---

#### [2 Básico] The design of the UNIX operating system.

*Bach, Maurice J.*  
*PTR Prentice Hall,, Englewood Cliffs, N. J : (1990)*  
 0132017997

---

**[3 Básico] The UNIX programming environment /**

*Brian W. Kernighan, Rob Pike.*

*Prentice Hall,, Englewood Cliffs (New Jersey) : (1984)*

*013937681X*

---

**[4 Básico] Understanding the Linux Kernel /**

*Daniel P. Bovet and Marco Cesati.*

*O'Reilly,, Beijing [etc.] : (2003) - (2nd ed.)*

*0-596-00213-0*

---

**[5 Básico] UNIX programación avanzada /**

*Francisco Manuel Márquez García.*

*Ra-Ma,, Madrid : (2004) - (3ª ed.)*

*84-7897-603-5*

---

**[6 Básico] Linux kernel internals /**

*Michael Beck...[et al.].*

*Addison-Wesley,, Harlow : (1998) - (2nd ed.)*

*0201331438*

---

**[7 Básico] Linux kernel programming :algorithms and structures of version 2.4 updated and extended /**

*Michael Beck,...[et.al.].*

*Addison-Wesley,, London : (2002) - (3rd. ed.)*

*0201719754*

---

**[8 Básico] The Linux Kernel book /**

*Rémy Card ; Eric Dumas ; Franck Mével.*

*John Wiley & Sons,, Chichester : (2000)*

*0471981419*

---

**[9 Básico] Linux Core Kernel commentary /**

*Scott Maxwell.*

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

*1576104699*

---

**[10 Recomendado] Advanced UNIX programming /**

*Mara J. Rochkind.*

*Prentice-Hall,, Englewood Cliffs, N. J : (1987)*

*0130118184*

---

**[11 Recomendado] PC interno 5 : programación de sistemas /**

*Michael Tischer, Bruno Jennrich.*

*Marcombo,, Barcelona : (1996)*

*8426710816*

---

**[12 Recomendado] Advanced programming in the Unix environment /**

*W. Richard Stevens , Stephen A. Rago.*

*Addison-Wesley,, Upper Saddle River, NJ : (2005) - (2nd ed.)*

*0-201-43307-9*

---

## Organización Docente de la Asignatura

Contenidos	Horas					Competencias y Objetivos
	HT	HP	HCT	HTT	HAI	
TEMA 1: INTRODUCCIÓN AL LINUX	2				1	Conocer una descripción de la arquitectura y estructura del núcleo
TEMA 2: INTERRUPCIONES	2				2	<p>Dominar el concepto de interrupción y los pasos que ejecuta el núcleo cuando se realiza una interrupción. Saber como implementa Linux el sistema de interrupciones, analizar las estructuras de datos y procedimientos que permiten gestionar y atender las interrupciones hardware. Conocer el mecanismo de interrupción software de las llamadas al sistema, sus pasos, estructuras y procedimientos. Conocer el sistema de señales, las estructuras y los procedimientos que gestionan el envío y recepción de señales</p>
TEMA 3: CARGA DEL NÚCLEO E INICIALIZACIÓN	1				1	<p>Saber los pasos que ocurren desde que se enciende el computador, hasta que este presenta el símbolo del procesador de comandos. Analizar y comprender el programa de inicialización init.</p>
TEMA 4: LLAMADAS AL SISTEMA	1				1	<p>Dominar el funcionamiento a nivel de programador de las principales llamadas al sistema, programar utilidades relativas a procesos, señales, manejo de ficheros, relativas a directorios, protección y manejo de tiempos.</p>

Contenidos	Horas					Competencias y Objetivos
	HT	HP	HCT	HTT	HAI	
TEMA 5: PROCESOS	4				4	Conocer los conceptos sobre procesos y su estructura interna. Saber su realización. Analizar las estructuras de datos para el manejo de procesos. Saber la implementación de estructuras y procedimientos para la gestión de CPU. Conocer las 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	5				5	Conocer las estructuras y procedimientos que asignan memoria virtual a un proceso. Analizar la técnica de paginación las estructuras de tablas de páginas y los procedimientos que las manejan. Saber los procedimientos que permiten tratar el disco como una prolongación de la memoria SWAP. Manejar las llamadas que permiten proyectar un fichero de disco en memoria, MMAP. Dominar las llamadas relacionadas con el manejo de memoria como BRK.

Contenidos	Horas					Competencias y Objetivos
	HT	HP	HCT	HTT	HAI	
TEMA 7: SISTEMA DE FICHEROS	5				5	Conocer 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. Saber uno de los sistemas de ficheros reales que soporta. Saber la implementación de las operaciones sobre ficheros, OPEN, CLOSE, READ, WRITE, MOUNT.
TEMA 8: COMUNICACIÓN ENTRE PROCESOS	5				5	Analizar y conocer las primitivas de comunicación entre procesos PIC. Saber los mecanismos de mensajes, semáforos, pipe y memoria compartida, con las estructuras de datos y procedimientos que los implementan. Conocer sockets.
TEMA 9: MANEJADORES DE DISPOSITIVOS. DRIVERS	5				5	Analizar y saber el funcionamiento de 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.

Contenidos	Horas					Competencias y Objetivos
	HT	HP	HCT	HTT	HAI	
PRÁCTICA 1: INSTALACIÓN DE LINUX		2			2	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.
PRÁCTICA 2: ALARMA		4			3	Conocer y manejar primitivas de tiempo real y reloj del sistema. Programar 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. Analizar y resolver el problema de la Sección Crítica.
PRÁCTICA 3: DISEÑO E IMPLEMENTACIÓN DE UN PROCESADOR DE COMANDOS		10			8	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. Utilización de las llamadas al sistema• FORK, WAIT, PIPE, EXEC. Diseñar a estructura de un Shell.
PRÁCTICA 4: MAPEAR UN FICHERO EN MEMORIA MMAP		2			1	Manejar los conceptos de proyectar un fichero en memoria. Saber utilizar a nivel de programa las llamadas que hacen posible el mapeo de un fichero en memoria, mmap.



Contenidos	Horas					Competencias y Objetivos
	HT	HP	HCT	HTT	HAI	
PRÁCTICA 5: COMPILAR EL NÚCLEO		4			3	Saber compilar el núcleo. Utilizar el comando Make. Diseñar tipos de compilación. Generar un nuevo Linux.
PRÁCTICA 6: CREAR UNA NUEVA LLAMADA AL SISTEMA		4			3	Diseñar una nueva llamada al sistema. Implementar dentro del núcleo la llamada.
PRÁCTICA 7: MEJORAR EL MANEJADOR DE MEMORIA		4			3	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.
Trabajo Tutorizado				15		Saber elaborar trabajos sobre la asignatura. Diseñar e implementar una memoria escrita. Saber expresar de forma oral, exponer y discutir estos trabajos en clase utilizando nuevas tecnologías informáticas. Saber trabajar en grupo. Aprender de manera autónoma con unas directrices.
Tutorías			5			Corresponden a cualquiera de las actividades teóricas, prácticas o trabajos.

## Equipo Docente

**SANTIAGO CANDELA SOLÁ**

**Categoría:** CATEDRÁTICO DE UNIVERSIDAD

**Departamento:** INFORMÁTICA Y SISTEMAS

**Teléfono:** 928458753 **Correo Electrónico:** [santiago.candela@ulpgc.es](mailto:santiago.candela@ulpgc.es)

Study of the internal structures of an operating system real, study of the functions and internal procedures of the kernel of real operating system. Study of subjects related to the last advances in operating systems.

Analysis and resolution synthesis of practical problems derivatives of a real implementation, using the calls and functions of system. Design and implementation of utilities of the system.

The course Design of the kernel of an Operating system is based on the study of Operating system LINUX developed by Linus Torvalds.

### SUBJECT 1: INTRODUCTION TO THE LINUX

Makes a description of architecture and structure of the kernel. Also one describes the steps that they follow in a call the system.

### SUBJECT 2: INTERRUPTIONS

The objectives try to strengthen interruption concept and the steps that the kernel executes when it is made an interruption. The mechanism of software interruption studies of calls to the system, their steps, structures and procedures, traps exceptions generated in the CPU. One studies of interruptions hardware, analyze the structures of data and procedures that they allow to manage and to take care of the interruptions hardware. Finally it studies signals, the structures and the procedures that send and receive signals.

### SUBJECT 3: START and BOOT KERNEL

The goals of the chapter deal with understanding the steps that happen since the computer power on, until this it present login and the shell symbol. To analyze and to study the boot program INIT.

### SUBJECT 4: SYSTEM CALLS

The chapter is related to the system calls, which they are supported by LINUX, becoming a functional study of each one of them and grouping them according to its application: to processes, signals, handling of files, relative to directories, protection and handling of times.

### SUBJECT 5: PROCESSES

The goals treat of studies the processes. It begins with a review to the concepts on processes, and it analyzes the structures of data for the handling of processes. To study structures and procedures for the CPU management. To study calls to the system that are related to the creation of processes FORK y CLONE, related with the execution of the image of a program EXEC, and completion of a process EXIT and the wait by a process WAIT.

### SUBJECT 6: MEMORY

To study the structures and procedures that assign virtual memory to process, analyzes the pagination technique, the structures tables of pages and the procedures that handle them. To study the procedures that to deal with the disc like a prolongation memory SWAP. To analyze the calls that allows projecting a disc file in memory, MMAP. It is ended up studying system calls related to the handling of memory like BRK.

### SUBJECT 7: ARCHIVES

To study virtual file system implemented by Linux, with its structures of buffer, the super block, inodo, and the structure of directory. To learn one real file system that it supports Linux. The implementation of the operations on files, CREAT, OPEN, CLOSE, READ, WRITE and SEEK, handling of directories and the implementation of calls, MOUNT, UMount, thus like other calls.

### SUBJECT 8: COMMUNICATION BETWEEN PROCESSES

The goals try to study primitive of communication between processes PIC. They study the mechanism of messages, pipe, and semaphores and shared memory, with the structures of data and procedures that implement them.

### SUBJECT 9: DRIVERS

To study several drivers classified by character: like the console, the keyboard, the printer; and block: like discs and CDrom.

### PROGRAM OF THE SUBJET

## PART 1: INTRODUCTION TO LINUX

### 1.1 HISTORY DE LINUX

### 1.2 LICENSE GNU A NEW PHILOSOPHY

### 1.3 KERNEL ARCHITECTURES

### 1.4 KERNEL STRUCTURE

### 1.5 CALLS SYNTHESSES AND USE

## PART 2: INTERRUPTIONS

### 2.1 INTRODUCTION

### 2.2 HARDWARE INTERRUPTIONS

### 2.3 SOFTWARE INTERRUPTIONS

### 2.4 SIGNALS

## PART 3: KERNEL LOAD AND BOOT

### 3.1 INTRODUCTION

### 3.2 KERNEL LOAD

### 3.3 BOOT THE INIT PROGRAM

### 3.4 THE MAIN PROGRAM

## PART 4: THE SYSTEM CALLS

### 4.1 INTRODUCTION

### 4.2 HANDLING OF PROCESSES

### 4.3 SIGNALS

### 4.4 HANDLING OF FILES

### 4.5 HANDLING OF DIRECTORIES AND SYSTEM OF FILES

### 4.6 PROTECTION

## PART 5: PROCESSES

### 5.1 CONCEPTS

### 5.2 TABLE OF PROCESSES

### 5.3 MANAGEMENT AND PLANNING OF PROCESSES

### 5.4 CREATION FORK, CLONE

### 5.5 EXECUTION EXEC

### 5.6 WAIT AND EXIT

## PART 6: MEMORY

### 6.1 PROCES MEMORY IN LINUX

### 6.2 VIRTUAL MEMORY OF A PROCESS

### 6.3 PAGINATION

### 6.4 INTERCHANGE OF PAGES WITH DISC SWAP

### 6.5 MAPPING OF FILE IN MEMORY MMAP

### 6.6 ALLOCATION MEMORY BRK

## PART 7: FILESYSTEM

### 7.1 VIRTUAL FILESYSTEM

### 7.2 BUFFER CACHES.

### 7.3 INODE TABLE.

### 7.4 SUPERBLOCK

### 7.5 FILE DESCRIPTOR.

### 7.6 OPEN AND CLOSE FILES

### 7.7 READ AND WRITE FILES

### 7.8 MOUNT A FILESYSTEM

## PART 8: PROCESSES COMMUNICATION

### 8.1 MESSAGES

### 8.2 SEMAPHORES

### 8.3 SHARED MEMORY

### 8.4 PIPE

### 8.5 SOCKETS

## PART 9: DEVICES DRIVERS

### 9.1 INPUT OUTPUT

### 9.2 CONSOLE

### 9.3 KEYBOARD

### 9.4 PRINTER

### 9.5 RAMDISK

### 9.6 DISCK

### 9.7 CDROM

## Final MARK

For the final mark we considers the practices, the works exposed in class, and the partial and/or final examinations. Other activities that we denominated “ others, like to have an attitude participative in class, good relationship, it can improve the note, so that the weight of each one of these aspects in the final mark, expressed in percentage is the following one:

40% Practices 20% Works 20% exam Part 1 20% exam Part 2 10% Others (To improve mark)