

## Sistemas Operativos Primera Práctica

### Introducción

LINUX suministra un intérprete de comandos shell que es necesario conocer para aprender a programar tareas específicas tanto para administrar el sistema como para realizar utilidades personalizadas.

Para poder desarrollar correctamente cualquiera de las prácticas que se proponen en la asignatura Sistemas Operativos a lo largo del curso académico, será imprescindible conocer la programación shell del sistema operativo Unix.

En esta primera práctica se pretende que el alumno tome conocimiento de la importancia de la shell de Linux así como de los temporizadores desde el punto de vista del diseño de los Sistemas Operativos, es habitual su utilización por parte de los sistemas operativos para el control de determinadas tareas: planificación de procesos, estadísticas del sistema, tiempos de espera para protocolos de red etc.

### Objetivo

Implementación de un MINI-SHELL, desarrollado en lenguaje shell y lenguaje C, sobre sistema operativo Linux que realice la función de intérprete de comandos. El programa deberá seguir las normas de diseño y cumplir los requisitos que se establecen en el enunciado.

Los usuarios que pueden acceder a **mini-shell** estarán almacenados en un fichero denominado **fusuarios**, donde aparecerán su **login, password** en el siguiente formato: **login:password**. Este fichero será proporcionado por los profesores de la asignatura y para operar correctamente se deberá comprobar que el fichero es correcto y el usuario introducido se encuentra dentro del fichero y la clave proporcionada es la correcta. Si los datos proporcionados no fueran correctos no se permitirá entrar al usuario en el mini-shell.

El programa mini-shell, deberá reconocer las órdenes que se indicarán a continuación y que deberán estar almacenadas inicialmente en un fichero de texto denominado **fordenes**, con el siguiente formato:

**nombre-orden:fecha\_hora\_ultimo\_acceso**

Donde:

Nombre-orden: Nombre de la orden que deberá introducir el usuario.

Fecha\_hora\_ultimo\_acceso: Última vez que se ejecutó la orden.

Nivel-acceso: Especifica los usuarios que podrán ejecutar dicha orden.

Si el usuario intenta ejecutar una orden que no se encuentra en este fichero se dará el siguiente mensaje de error: **“orden no encontrada”**.

Todos los ficheros de configuración (fusuarios, fordenes), deberán estar almacenados en el subdirectorio **cfg**, dentro del directorio de la práctica.

Del programa **mini-shell** se saldrá al ejecutar la orden **salir**, permitiendo devolver el control al shell del sistema.

Cuando el programa mini-shell se inicia deberá comprobar que existen todos los ficheros de configuración, en caso contrario el programa mostrará “Faltan parámetros de configuración” y saldrá del programa. Si todos los

## Curso 2008-2009

ficheros de configuración están presentes, pedirá el login y password del usuario, si el usuario introducido o su clave fuera incorrecta, no se podrá iniciar sesión. Como máximo se permitirán 3 intentos de inicio de sesión incorrectos. Si el inicio de sesión es correcto se mostrará como prompt de nuestro mini-shell, el login del usuario que ha iniciado sesión seguido del carácter @ y a continuación la cadena **minishell\$**, y esperará a que el usuario le envíe órdenes a ejecutar.

Ejemplo: si el usuario que ha iniciado sesión es administrador, en la línea de órdenes aparecerá:

**administrador@minishell\$**

Cada orden ejecutada, correcta o incorrectamente deberá ser almacenada en un fichero histórico, denominado fhistorico y que deberá permitir como mucho 50 órdenes, si el número de líneas supera este límite se borrará la orden más antigua, para que se pueda almacenar una más reciente.

Las órdenes que deberá reconocer el mini-shell son:

### - **arbol\_directorio nombre\_directorio**

Esta orden permitirá mostrar el contenido completo del directorio que recibe como parámetro incluyendo los subdirectorios que contenga. Se han de realizar todos los controles de error necesarios para que al usuario no le aparezca ningún mensaje del sistema (Si es un directorio, existencia, etc.). La orden será implementada en lenguaje de programación Shell. La información deberá mostrarse formateada adecuadamente, de tal forma que se vea de forma correcta.

Ejemplo: arbol\_directorio prueba

- prueba
  - f1
  - f2
  - dir1
    - f11
    - f12
    - dir2
      - f21
      - f22
  - dir2
    - f3
    - f4
  - f5

- **cambia\_clave:** Permitirá cambiar la palabra clave del usuario que se especifique como parámetro. La nueva clave se deberá pedir por duplicado y comprobar que son la misma. Se deberán realizar todos los controles necesarios para que al usuario no le aparezca ningún mensaje del sistema (introducción de clave vacía, no existencia del usuario, etc.). La implementación se realizará en lenguaje de programación Shell.

Formato: **cambia\_clave usuario**

**Petición de clave anterior: claveanterior**  
**Nueva clave: clavenueva**

### Repetir nueva clave: clavenueva

- **historico\_ordenes:** Deberá mostrar del fichero histórico tantas líneas como se especifiquen en el parámetro que recibe. Las órdenes que muestre serán las más recientemente ejecutadas. La orden será implementada en lenguaje Shell. Se deberán realizar todos los controles necesarios para que al usuario no le aparezca ningún mensaje del sistema.

**Formato: historico\_ordenes num\_linea**

- **ejecutar-temporizador intervalo ocurrencias accion param1 param2 ...**

Esta orden permitirá ejecutar un temporizador con la configuración pasada como parámetro en un nuevo terminal xterm, mientras que en el terminal anterior se podrán seguir ejecutando otras órdenes. La orden llamará a un programa realizado en lenguaje C llamado “**temporiza**” y que recibirá como parámetros de entrada los parámetros que reciba la orden. Se deberán realizar todos los controles necesarios para que al usuario no le aparezca ningún mensaje del sistema.

**Formato: ejecutar-temporizador intervalo ocurrencias accion param1 param2..**

Donde:

Intervalo: el número de segundos que transcurrirá entre cada ocurrencia de ejecución del temporizador.

Ocurrencias: N° de veces que se va a ejecutar el temporizador antes de pasar a la ejecución de otro.

Acción: Tarea que va a realizar, valores numéricos del 1 al 5 que pertenecen a:

1. Mostrar los grupos de usuarios existentes en el sistema, debidamente formateados y los usuarios que pertenecen a dicho grupo(fichero /etc/group).
2. Localizar un fichero pasado como parámetro y mostrar las ubicaciones donde se ha encontrado (Comando locate).
3. Mostrar si un fichero pasado como primer parámetro existe en un directorio pasado como segundo parámetro.
4. Ordenación de todos y cada uno de los ficheros enviados como parámetros, hasta un máximo de 3, almacenando la versión ordenada en un directorio denominado “**version\_ordenada**”.  
Uso del comando sort.
5. Comprobar si en el fichero de usuarios del sistema (/etc/passwd), existen los usuarios que se pasen como parámetros (como máximo 3). Uso del comandos grep, cut.

Parámetro: será necesario para aquellas acciones que necesiten adicionalmente un parámetro adicional

- **Iniciar :** Orden que permitirá iniciar el minishell con un usuario distinto al actual, realizando todas las comprobaciones de error necesarias.

**Formato: iniciar login\_usuario**

- **Salir**

Devolverá el control al shell del sistema.

La documentación se presentará en formato digital y según se especifica en las normas de entrega disponible en el campus virtual de la asignatura. No se aceptará ninguna práctica que no cumpla los requisitos aquí especificados.