A terminal window with a white title bar containing three window control buttons (minimize, maximize, close). The main area has a black background. A yellow rectangular box contains the title text in a pixelated font. Below it, a blue rectangular box contains the authors' names in a white sans-serif font. At the bottom, the text 'Bitácora - Semana 7' is displayed in a white pixelated font. A vertical scrollbar is on the right side of the terminal window.

Administración de Sistemas Unix/Linux

Bonilla Reyes Dafne - Cabrera Ramírez Carlos - García Ponce José Camilo
García Sánchez Abigail

Bitácora - Semana 7

Tabla de Contenido

01 Conceptos
nuevos

02 Conceptos de
repaso

03 Máximas

04 Datos
adicionales



01

Conceptos nuevos

Semana 7

Conceptos



Ligas

Las ligas (también llamadas enlaces o links) en Linux son como accesos directos en Windows.



Procesos con fork y exec

Los procesos se pueden crear con fork y exec.

- fork crea un clon de un proceso pero el clon tiene un identificador diferente y permite crear varios init.
- exec ejecuta un comando desde el bash, no crea un nuevo proceso solo reemplaza el bash con el comando a ejecutar.

Conceptos



init

Fue reemplazado por systemd. Usa scripts sh para inicializar los servicios de manera secuencial.



Proceso zombie

Es un proceso que ya no se ejecuta pero sigue en la tabla de procesos.



nice

Fue reemplazado por systemd. Usa scripts sh para inicializar los servicios de manera secuencial.

Conceptos



nice

nice es un valor que determina la prioridad de un proceso en la CPU.

Un proceso con un valor nice más alto tiene menos prioridad, mientras que uno con un valor nice más bajo (más cercano a -20) tiene más prioridad en la asignación de tiempo de CPU.

Conceptos

<code>grub.d</code>	Contiene script ejecutables
<code>profile.d</code>	Contienen scripts que se ejecutan cada vez que iniciamos un bash
<code>.bashrc</code>	Contiene funciones y alias que pueden usarse en el sistema entero para el usuario



DoubleSpace

Duplica el espacio al comprimir la información



TestDisk

Hace un escaneo rápido y luego otro donde se pueden buscar archivos perdidos

Comandos

<code>set</code>	Permite crear y cambiar variables de ambiente, así como opciones del shell
<code>nohup</code>	Crea un proceso independiente al que se está ejecutando en el momento
<code>W</code>	Permite mostrar a los usuarios logeados en el sistema, así como datos sobre ellos
<code>renice -n 10 -p 1454</code>	Se cambia el nice del proceso con PID 1454 a un nice de 10

Actividad de clase



Cantidad de archivos en /dev

En la clase del lunes vimos como obtener la información de la cantidad de archivos de cada tipo que están en el directorio /dev usando comandos de Linux. Por ejemplo:

- Los archivos regulares tienen “-” antes de los permisos
- Los directorios tienen “d”
- Los enlaces simbólicos tienen “l”
- Y los dispositivos de caracteres tienen “c”

También el martes vimos un script para obtener la cantidad de cada tipo de archivos en un directorio.

Actividad del lunes para la bitácora

```
#!/bin/bash
# Verifica si se paso un argumento
if [ -z "$1" ]; then
    echo "Uso: $0 <directorio>"
    exit 1
fi
# Verifica si el argumento es un directorio valido
if [ ! -d "$1" ]; then
    echo "Error: '$1' no es un directorio valido o no existe"
    exit 1
fi
# Cuenta los tipos de archivos en el directorio dado
info=$(ls -l "$1" | cut -c1 | sort | uniq -c)
# Mostrar la informacion diciendo que tipo de archivo es
echo "$info" | while read line; do
    count=$(echo $line | awk '{print $1}')
    type=$(echo $line | awk '{print $2}')
    case $type in
        "-") echo "Archivos regulares: $count";;
        "d") echo "Directorios: $count";;
        "c") echo "Dispositivos de caracteres: $count";;
        "b") echo "Dispositivos de bloques: $count";;
        "l") echo "Enlaces simbolicos: $count";;
        "s") echo "Sockets: $count";;
        "p") echo "Pipes: $count";;
    esac
done
```

Por ejemplo:

```
"WATCHER ALL = (erick)CMDWATCHER"
```

Los miembros de WATCHER pueden ejecutar los comandos de CMDWATCHER en todas las computadoras pero como erick.

Directorios

mapper

Tienen volúmenes lógicos del sistema que administra LVM

dev/cpu/

Tiene tantos directorios dentro como núcleos en el equipo.



Aprendizaje
adicional



`java.sh`

En `java.sh` hay direcciones para que java pueda funcionar como Java Home

`Export en .sh`

Los `export` en `.sh` hace que las variables definidas las “puedan ver” los demás procesos

`Archivos de profile.d`

Los archivos de `profile.d` se ejecutan con `profile` (es un archivo de ASCII)



`/dev/null`

Es un archivo especial que actúa como “hoyo negro” para los datos

Terminal por default

Por default en Linux se usa la terminal número 7, es decir, la TTY7, que es donde se ejecuta el entorno gráfico (Xorg/Wayland)

Init vs systemd

init ejecuta procesos secuencialmente y usa runlevels.
systemd inicia procesos en paralelo y usa unidades (units).
systemd es más rápido y el estándar en la mayoría de distribuciones.

02

Conceptos de repaso

Semana 7





Datos de repaso

systemd es un sistema de inicio moderno que reemplaza a init, optimizando el arranque de Linux.

Ejecuta procesos en paralelo, usa unidades (units) en lugar de runlevels y gestiona servicios con systemctl.

También incluye journald para registros y mejora la administración de dependencias entre servicios.



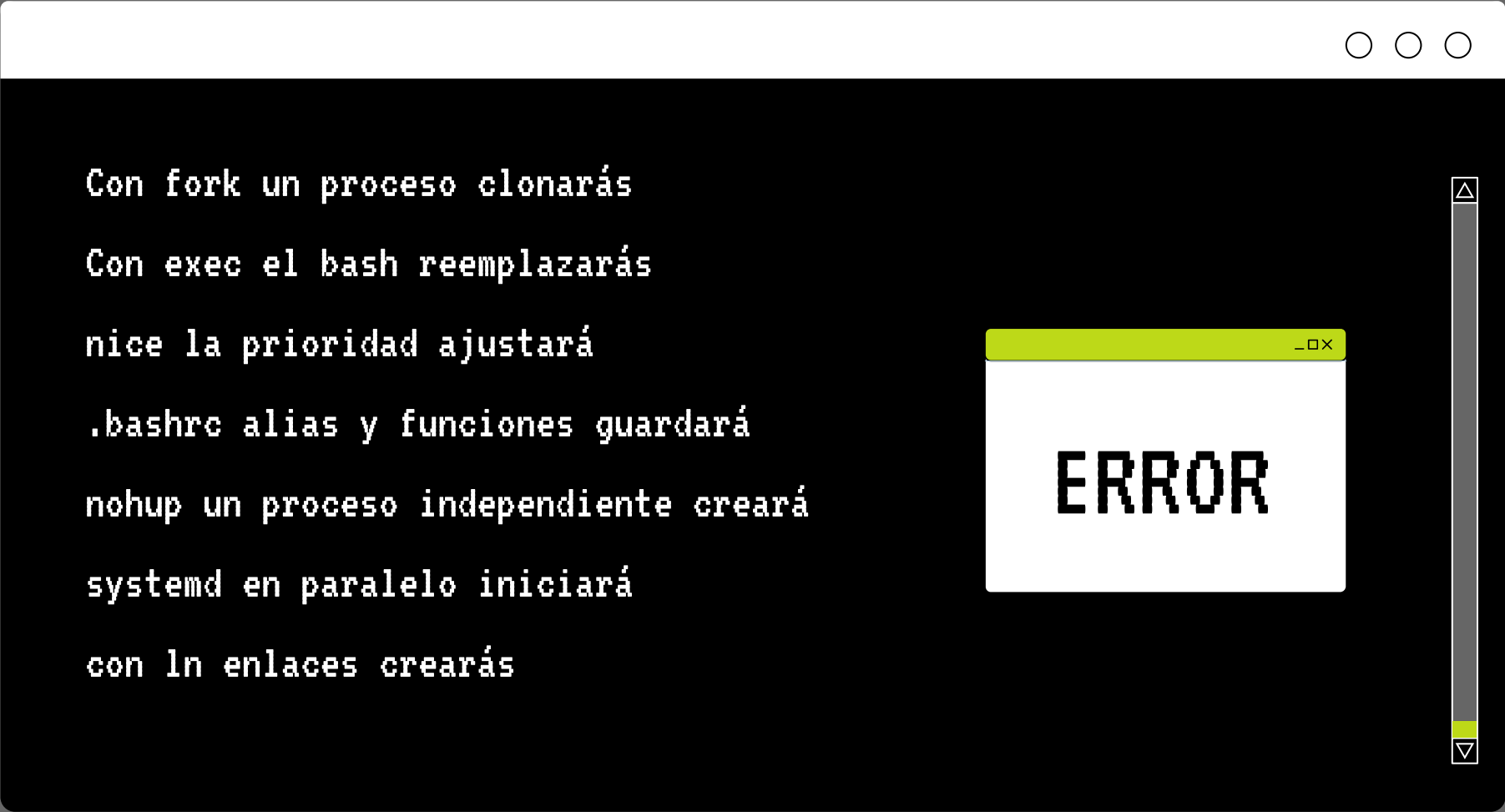


03

Máximas

Semana 7





A terminal window with a black background and white text. The text is a list of Linux process management commands. To the right of the text is a small white window with a yellow title bar and the word 'ERROR' in large black letters. On the far right is a vertical scrollbar with a yellow highlight at the bottom.

Con fork un proceso clonarás
Con exec el bash reemplazarás
nice la prioridad ajustará
.bashrc alias y funciones guardará
nohup un proceso independiente creará
systemd en paralelo iniciará
con ln enlaces crearás

ERROR

04

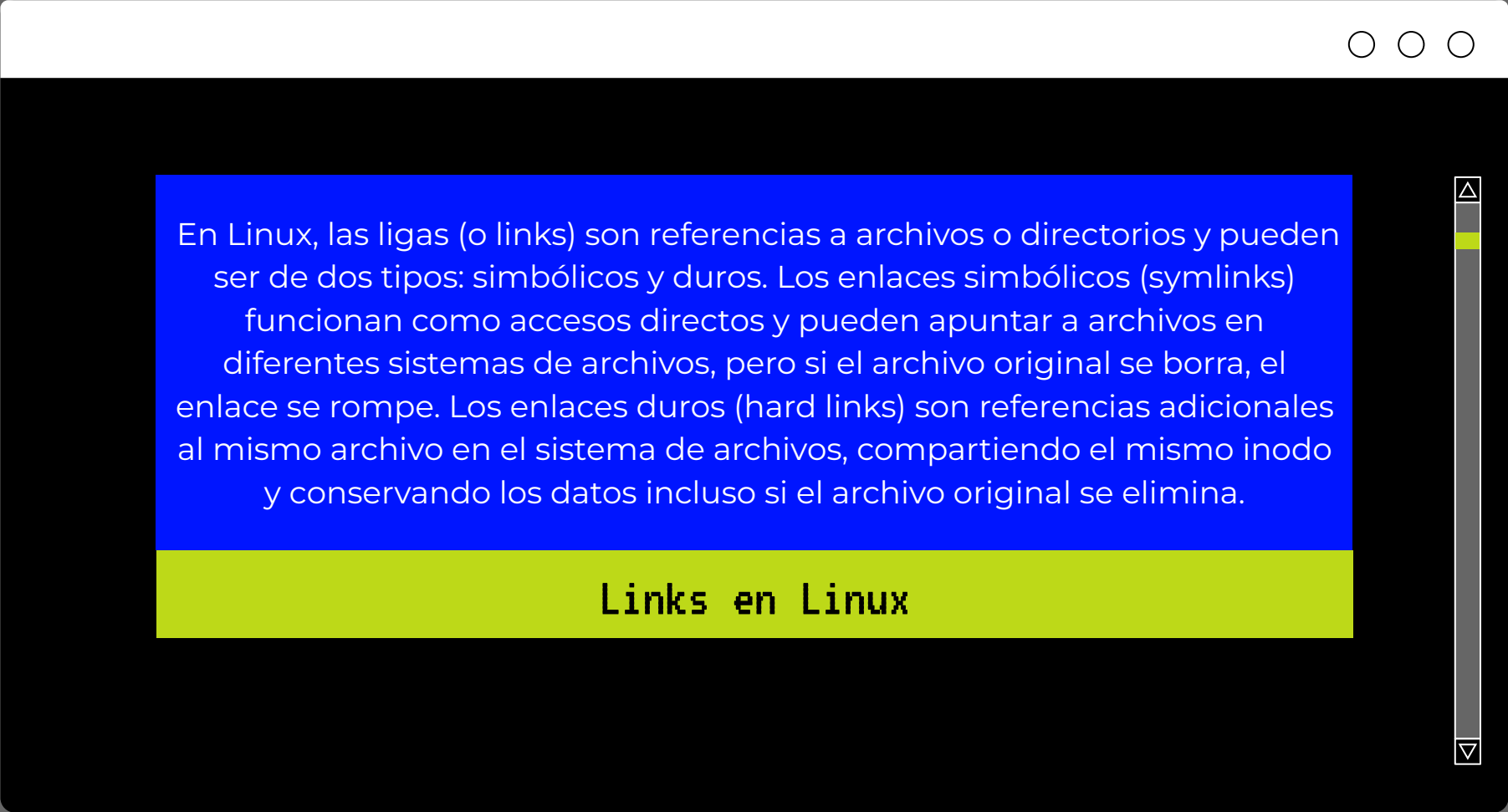
Datos adicionales

Semana 7





Investigación Extra



En Linux, las ligas (o links) son referencias a archivos o directorios y pueden ser de dos tipos: simbólicos y duros. Los enlaces simbólicos (symlinks) funcionan como accesos directos y pueden apuntar a archivos en diferentes sistemas de archivos, pero si el archivo original se borra, el enlace se rompe. Los enlaces duros (hard links) son referencias adicionales al mismo archivo en el sistema de archivos, compartiendo el mismo inodo y conservando los datos incluso si el archivo original se elimina.

Links en Linux



En systemd, device units (.device) permiten representar dispositivos de hardware como servicios administrables dentro del sistema. Esto significa que systemd puede detectar, gestionar y establecer dependencias entre dispositivos y otros servicios, asegurando que ciertos servicios solo se inicien cuando el hardware requerido esté disponible.

.device



tty (TeleTYpewriter) se refiere a las terminales en Linux y Unix, tanto físicas como virtuales.

Terminales físicas → Consolas accesibles con Ctrl + Alt + F1 a F6 (ejemplo: /dev/tty1).

Terminales virtuales (PTY - Pseudo Terminals) → Consolas dentro de un entorno gráfico o por SSH (ejemplo: /dev/pts/1).

tty

Parte de cada integrante

Dafne Bonilla Reyes	Presentación y diseño
Carlos Cabrera Ramírez	Presentación y aportaciones técnicas
José Camilo García Ponce	Resumen de la semana
Abigail García Sánchez	Presentación e investigación

