

Administración de Sistemas Unix/Linux

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

Bitácora - Semana 8

Tabla de Contenido

01 Conceptos
nuevos

02 Conceptos de
repaso

03 Máximas

04 Datos
adicionales





01

Conceptos nuevos

Semana 8



Conceptos



Procesos

Un proceso puede:

- Estar listo para ejecutar y en memoria
- Listo para ejecutar y descargado (aún no se carga en memoria)
- Dormido
- Descargado
- Creado
- Zombie

Los procesos tienen modo kernel y modo usuario

Conceptos



Primer plano

Primer plano es la forma por defecto en la que se ejecuta un proceso del usuario. Esto significa que no se puede hacer algo hasta que termine.



Segundo plano

Los procesos se ejecutan y permiten hacer otras cosas. Con '&' al final del comando podemos ponerlo en segundo plano o usando Ctrl + Z



Niveles de ejecución (runlevels)

Cada nivel de ejecución dice como se debe comporta la computadora y que servicios tener.

Originalmente fueron utilizados en sistemas basados en System V init, pero con la adopción de systemd, los runlevels fueron reemplazados por targets, que ofrecen mayor flexibilidad y control.

0	Computadora apagada
1	Mono usuario
2 - 5	Multiusuario
6	Sistema reiniciando

Comandos

<code>Ctrl + z</code>	Suspende un proceso
<code>jobs</code>	Muestra los comandos suspendidos
<code>fg</code>	Reanuda el último proceso suspendido. También, con 'fg %<num>' se reanuda un proceso en específico
<code>nice</code>	Inicia un proceso con un nice en específico. Por ejemplo, 'nice -n <#nice> <comando>'

Archivos

`/etc/shells`

Se definen los distintos shells que se pueden usar.

Directorios

`/etc/alternatives`

Hay varios enlaces que sirven para ver cual es la versión de varios programas y las versiones que se usarán por default. También apuntas a las versiones de los programas instalados y son útiles para poder cambiar la versión o implementación de algunos programas sin necesidad de hacer cambio a rutas del sistema.



Aprendizaje
adicional



initd

initd funciona mediante niveles de ejecución (además de usar scripts de sh ejecutandolos secuencialmente) y systemd mediante unidades

Boot Device

Cuando una computadora no encuentra el Boot Device es posible que se encuentre dañada.





Diferentes Shells

Bash: Es el shell predeterminado en varias distros (como Ubuntu y Fedora). Basado en Bourne Shell (sh), soporta arreglos, aritmética y expansión avanzada, es POSIX compliant.

Zsh: Es similar a bash y permite mayor personalización, provee soporte a diferentes frameworks para hacer tareas más complejas. Esto hace que sea más lento para inicializarse.

Dash: Es ligera y rápida, diseñada para scripts POSIX. No provee características tan avanzadas como bash/zsh.

Csh: Tiene una sintaxis inspirada en C, pero no provee una gran compatibilidad con POSIX. Implementa características avanzadas como alias y job control.



02

Conceptos de repaso

Semana 8





Datos de repaso

Los procesos tienen modo kernel y modo usuario.
Información adicional de los procesos y cómo crearlos.
Repaso sobre init.d y cómo inicia.
Continuación del comando nice y cómo cambiar el nice de los procesos.





03

Máximas

Semana 8



Ctrl + Z un proceso suspenderá
Con fg al primer plano volverá
jobs los procesos suspendidos listará
nice la prioridad cambiará
/etc/shells los shells disponibles guardará
systemd los runlevels reemplazará
Con Zsh más personalización tendrás



ERROR

04

Datos adicionales

Semana 8



Actividades realizadas en clase

Comparación de los shells: zsh, bash, dash y csh.

Bash: es el shell predeterminado en varias distros como Ubuntu y Fedora, es basado en Bourne Shell (sh), soporta arreglos, aritmética y expansión avanzada, compatible con POSIX, es fácil de usar, es eficiente y bueno en sistemas sin tantos recursos

Zsh: similar a Bash pero soporta más personalización, soporta diferentes frameworks para hacer cosas más complejas, soporta corrección automática de comandos, tiene buen rendimiento en algunas tareas, puede ser algo lento para arrancar por sus características adicionales

Actividades realizadas en clase

Dash: es ligera y rápida, diseñada para scripts POSIX, no tiene características tan avanzadas como Bash o Zsh, muy útil para ejecutar scripts de inicio con bajo consumo de recursos

Csh: sintaxis inspirada en C, no es tan compatible con POSIX, tiene características avanzadas como alias y job control, Tcsh es la versión mejorada y con más características (como edición de línea de comandos), no es tan popular por su mala compatibilidad

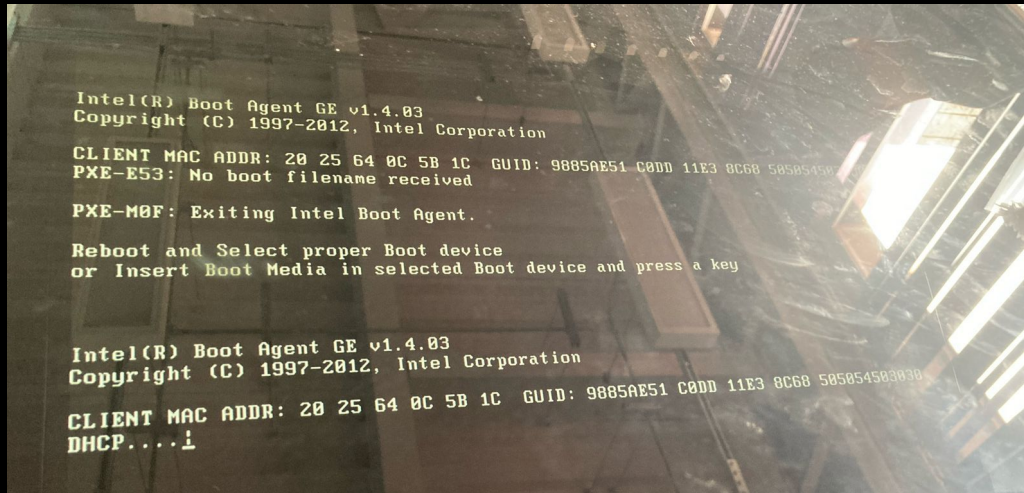
Dependiendo de lo que busquemos es el que debemos elegir, por ejemplo para scripts de POSIX es bueno Dash (es rápido), para uso diario es bueno Bash y Zsh (tiene gran personalización)

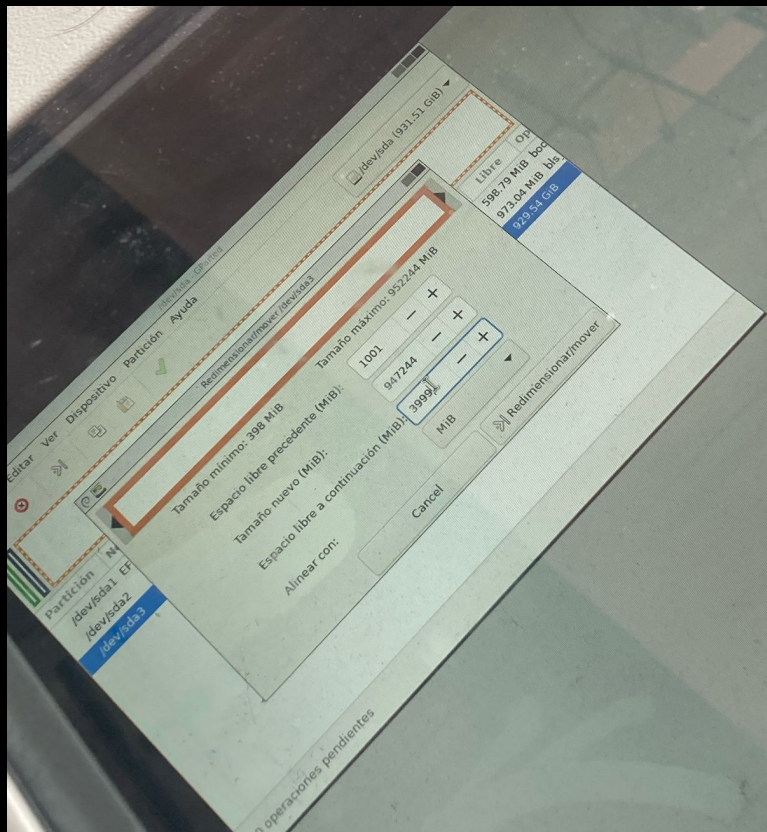
Actividades realizadas en clase

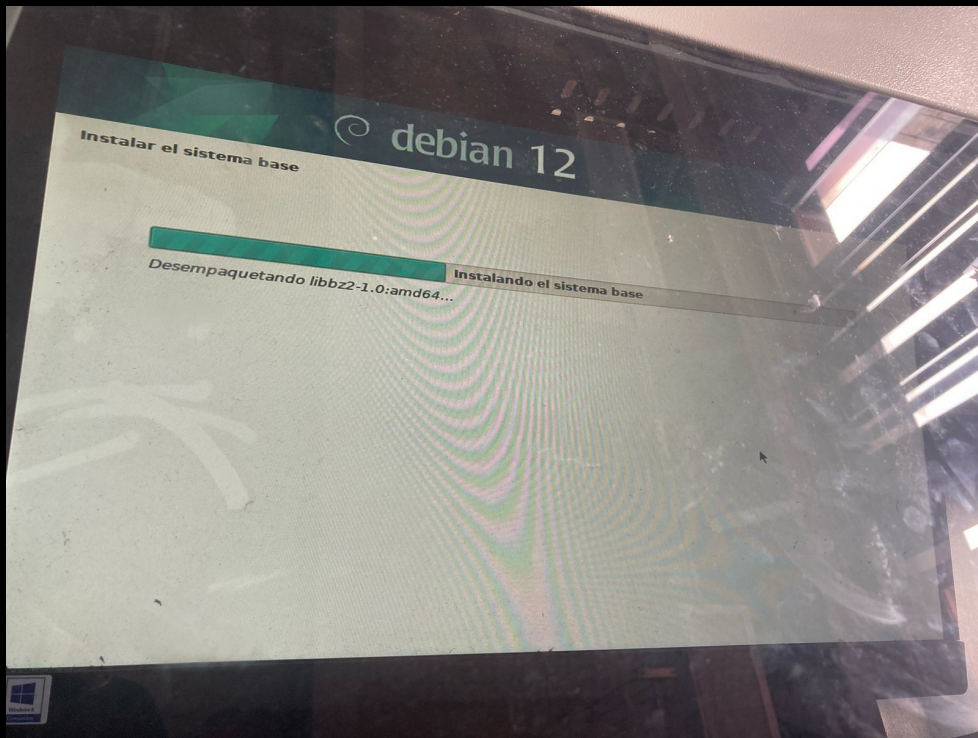
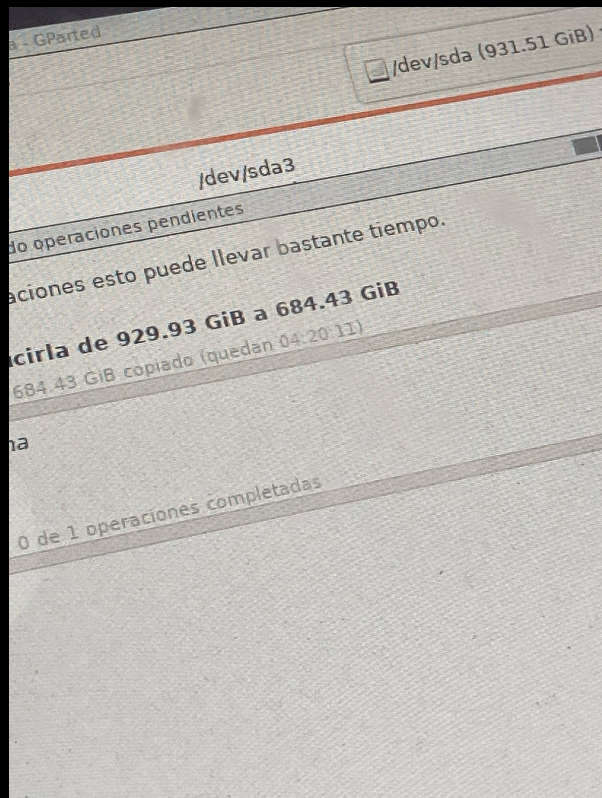
Features	Bash	Zsh	Fish	Tcsh	Ksh	Ash	Sh
Bourne shell compatible	✓	✗	✗	✗	✓	✓	✓
Command history	✓	✓	✓	✓	✓	✓	✓
Unicode support	✓	✓	✓	✓	✓	✗	✓
Command name completion	✓	✓	✓	✓	✓	✗	✗
Path completion	✓	✓	✓	✓	✓	✗	✓
Command arg completion	✓*	✓*	✓*	✓*	✗	✗	✗
Wildcard completion	✓	✓	✓	✗	✗	✗	✗
Automatic suggestion	✗	✓*	✓	✗	✗	✗	✗
Colored directory listings	✓	✓	✓	✓	✓	✓	✓
Text highlighting	✓	✓	✓	✓	✓	✓	✓
Syntax highlighting	✗	✓*	✓	✗	✗	✗	✗
Auto correction	✗	✓	✓	✗	✗	✗	✗
User-defined function	✓	✓	✓	✗	✗	✓	✓
Exception handling	✓	✓	✓	✗	✓	✓	✓
Search and replace	✓	✓	✓	✓	✓	✗	✗
Arithmetic	✓	✓	✓	✓	✓	✓	✓
Floating point support	✗	✓	✓	✗	✓	✗	✗
Linear array support	✓	✓	✓	✓	✓	✗	✗
Associative array support	✓	✓	✗	✗	✓	✗	✗
Built-in pattern matching	✓	✓	✓	✓	✓	✗	✗
Subshells	✓	✓	✗	✓	✓	✓	✓

Actividades realizadas en clase

Intentos por revivir las computadoras del laboratorio



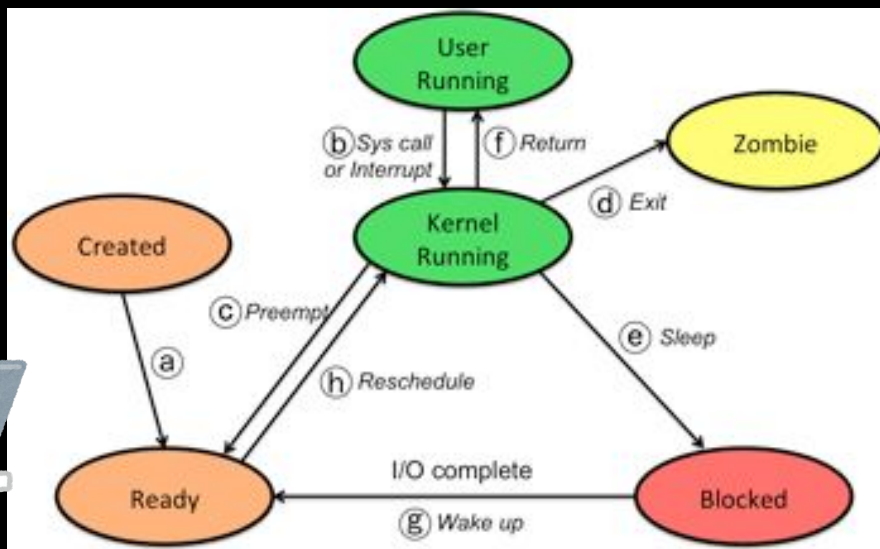
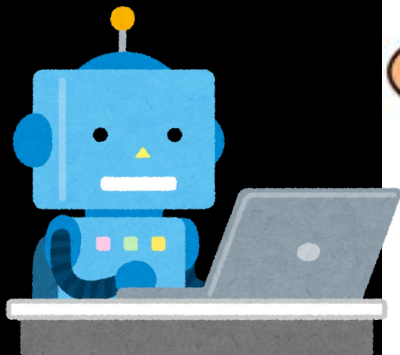






Investigación Extra

Diagrama de estados de los procesos en Linux



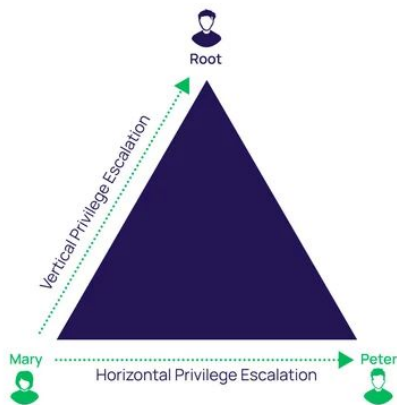
Niveles de ejecución



Modo ejecución

Finalidad

0	Alto
1	Modo de usuario único (Monousuario)
2	Multiusuario
3	Multiusuario con soporte de red
4	Multiusuario con soporte de red
5	Multiusuario gráfico (X11)
6	Reinicio



La elevación de privilegios es el proceso de obtener permisos más altos o root (administrador) en un sistema operativo. En sistemas basados en Unix, esto es crucial para ejecutar comandos que afectan al sistema o a otros usuarios.

Elevación de privilegios

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

