TRABAJO FINAL
SISTEMAS OPERATIVOS

Consideraciones Generales

Como parte de los requisitos para obtener la cursada de la materia, es necesario realizar un trabajo grupal de investigación de algunos de los temas propuestos a continuación.

Los trabajos finales deberán cumplir con las siguientes condiciones para su aprobación:

- Elaboración de **un documento**, que se entregue **impreso y en carpeta**. Dicho documento deberá contener una **parte teórica y otra práctica** (capturas, trazas, demostraciones de uso, etc.) correspondientes a cada temática, además de las **conclusiones** alcanzadas por el grupo a cargo de la investigación. Se deberá **consignar toda la bibliografía** empleada para el desarrollo del TP, además de referencias a los sitios web utilizados.
- **Elaboración de diapositivas** para presentar lo investigado en una exposición frente a la clase.
- **Entrega en CD/DVD** del **Documento**, las **Diapositivas**, y alguna **parte práctica, con instructivo de uso.** Puede ser un programa o una Máquina Virtual. Dependerá de cada caso.
- Presentación del tema ante la cátedra y el resto de los compañeros.

Propuestas de trabajos a resolver

1. Debian vs. RedHat

Una distribución Linux (distro) es una distribución de software basada en el núcleo Linux que incluye determinados paquetes de software para satisfacer las necesidades de los usuarios finales. Están compuestas de software libre, aunque a menudo incorporan software propietario.
Además del núcleo Linux, las distribuciones incluyen habitualmente las bibliotecas y herramientas del proyecto GNU y el sistema de ventanas X Window System.
Dentro de las distribuciones pioneras, se destacan Debian y RedHat.

Realizar una comparación entre dos distribuciones, una de la rama de Debian y otra de la rama de RedHat (recomendamos CentOS). El análisis debe ser detallado y la comparación debe incluir todos los aspectos tanto de un sistema operativo como de una distribución.
Como parte de la entrega deberá considerar la confección de una máquina virtual de las distros elegidas. Elaborar un informe y una exposición sobre ambas distribuciones especificando los siguientes detalles:

- Cómo se organiza un sistema GNU/Linux
- Qué es una distribución y cuál es su arquitectura, su estructura y sus componentes
- Aspectos técnicos de cada distro, tales como arquitectura hardware soportada, versión de kernel, sistemas de archivos soportados, gestores de ventanas compatibles, etc.
- Características de cada distribución: forma de su ciclo de vida, paquetes disponibles, formato de repositorios utilizados, gestor de paquetes, formato de paquetes, etc.
- Mecanismos de seguridad implementados.
- Ventajas y desventajas de uso de cada distro.

Construir ejemplos y/o casos que permitan mostrar en forma práctica los conceptos previamente expuestos.

Algunas páginas útiles:
https://www.debian.org/
https://www.debian.org/doc/manuals/debian-handbook/
https://www.redhat.com
https://www.centos.org/
https://es.wikipedia.org/wiki/Distribuci%C3%B3n_Linux#/media/File:Linux_Distribution_Timeline.svg

2. Cron y Anacron para tareas programadas

Cron y Anacron son herramientas utilizadas para la ejecución programada de tareas en Linux, de forma síncrona y asíncrona respectivamente. Ambas herramientas se encuentran incluidas por defecto en la mayoría de las distribuciones GNU/Linux.
Si bien Cron y Anacron permiten agendar tareas, su funcionamiento no es igual y en lugar de ser productos alternativos, son necesariamente complementarios.

Realizar una investigación en la que se analice el funcionamiento de Cron y Anacron para la ejecución de tareas en Linux. Explicar en forma detallada en qué consiste cada herramienta, cómo funcionan, cuáles son sus posibles configuraciones y cómo y dónde realizar las mismas.
Como parte de la entrega deberá considerar la confección de scripts en bash que puedan agendarse como tareas programadas para verificar su correcto funcionamiento, así como todos los archivos de configuración que se vean involucrados en cada una de las herramientas. La producción debe contemplar:

- Qué es Cron y qué es Anacron, para que se utilizan y en qué se diferencian
- Cuál es la forma en que se configuran las tareas programadas en cada herramienta
- Cuáles son los archivos de configuración involucrados y cuál es la función de cada uno.
- De qué forma el sistema operativo garantiza la ejecución de acuerdo a lo planificado.
- Análisis de bitácora o logs de ambas herramientas
- Mecanismos de seguridad empleados por ambas herramientas
- Ventajas y desventajas de uso.

Construir ejemplos que permitan mostrar en forma práctica los conceptos previamente expuestos.
Valerse de scripts en bash para la codificación de tareas agendadas.

Algunas páginas útiles:
http://man7.org/linux/man-pages/man8/cron.8.html
http://man7.org/linux/man-pages/man5/crontab.5.html
http://man7.org/linux/man-pages/man8/anacron.8.html

3. Rsyslog

Un archivo de registro (bitácora o log) es un registro de eventos que suceden durante un rango de tiempo. Mediante ellos, aplicaciones o dispositivos registran datos o información sobre quién, qué, cuándo, dónde y por qué ocurre un evento determinado. Los archivos de registro son útiles a la hora de detectar y diagnosticar fallos e intentar solucionarlos. Linux ofrece herramientas como Syslog y Rsyslog para gestionar los logs del sistema.
Rsyslog implementa el protocolo básico de syslog y lo extiende añadiendo funcionalidades y permitiendo el registro de logs generados en hosts remotos.
Realizar un informe detallado que explique cómo funciona el mecanismo de logs del sistema. Utilizar Rsyslog como herramienta base. El informe debe contemplar:

- Funcionamiento y finalidad del protocolo Syslog
- Estructura de un mensaje de log:
    - Prioridad (recurso y severidad)
    - Cabecera
    - Texto
- Demonio o demonios encargados de la recolección de logs.
- Cuáles son los archivos de configuración involucrados y cuál es la función de cada uno.
- Configuraciones requeridas para el funcionamiento de Rsyslog
- Herramientas para el análisis de logs.
- Ventajas y desventajas de uso.

Construir ejemplos que permitan mostrar en forma práctica los conceptos previamente expuestos.

Algunas páginas útiles:
https://debian-handbook.info/browse/es-ES/stable/sect.syslog.html
https://es.wikipedia.org/wiki/Syslog
http://www.makeinstall.es/2011/05/gestion-de-archivos-de-registro-del.html
https://www.rsyslog.com/

4. Sudo y Sudoers

El programa sudo (del inglés super user do) es una utilidad de los sistemas operativos tipo Unix, como Linux, BSD, o Mac OS X, que permite a los usuarios ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) de manera segura, convirtiéndose así temporalmente en superusuario.
En los sistemas derivados de Unix, existe un archivo llamado “sudoers” que controla los niveles del sistema de permisos. Allí, se permite o se deniega a los usuarios la posibilidad de obtener acceso de superusuario para la ejecución de ciertas tareas.
Realizar un informe detallado que explique cómo funciona el mecanismo de elevación de privilegios de un usuario regular. El informe debe contemplar:

- Qué tipos de operaciones requieren privilegios de root y por qué
- Diferencias entre la elevación de privilegios mediante los comandos “su” y “sudo”
- Funciones y características de la utilidad “sudo”
- Archivo sudoers:
    - Cómo añadir un usuario a sudoers
    - Sintaxis para la edición del archivo
    - Posibilidades para la elevación de privilegios
- Análisis de bitácora o logs
- Mecanismos de seguridad empleados.

Construir ejemplos que permitan mostrar en forma práctica los conceptos previamente expuestos.
Valerse de scripts en bash para la ejecución de procesos mediante la elevación de privilegios.

Algunas páginas útiles:
https://www.sudo.ws/
https://www.sudo.ws/man/1.8.27/sudoers.man.html
https://www.poftut.com/what-is-sudo-command-in-linux/

5. Minix

Minix es un sistema operativo surgido como clon de Unix. Fue desarrollado en la década del 80 por el Profesor Andrew Tanembaum. Es un software de código abierto, publicado bajo la Licencia GNU Lesser General Public License (LGPL). Debido a su gran valor en ámbitos académicos el sistema perdura en el tiempo y sirve como caso de estudio en muchas universidades del mundo.
Realizar un informe y una exposición que detalle todas las características funcionales de Minix, como sistema operativo. El informe debe considerar:

- Arquitectura/Estructura/Componentes del sistema
- Características técnicas de su kernel
- Programación en Minix
- Llamadas al sistema en Minix
- Procesadores compatibles con el sistema
- Administración de memoria

Mostrar en ejecución el sistema Minix y efectuar un programa sencillo en él.

Página oficial de Minix:
https://www.minix3.org/

6. Docker y Podman

Docker es un proyecto de código abierto que permite la automatización del despliegue de aplicaciones dentro de contenedores de software. De este modo, proporciona una capa adicional de abstracción, mediante la virtualización de aplicaciones en múltiples sistemas operativos.
Por otro lado Podman es un motor de contenedores (container engine) que puede ser un sustituto para Docker. Este motor ha sido desarrollado por la compañía Red Hat.

Realizar un informe y una exposición sobre estas tecnologías, especificando los siguientes detalles:

- Qué son y para que se utilizan
- Arquitectura/Estructura/Componentes (Engine, Image, Container)
- Ventajas y desventajas de uso. Comparativa con otras tecnologías (Vagrant, HyperV)
- Diferencias con virtualización completa
- Herramientas de Docker.
- Herramientas de Podman
- Comparativa entre ambas tecnologías

Construir ejemplos que permitan mostrar en forma práctica los conceptos previamente expuestos:

- Demostrar cómo se aplican los conceptos de SO (procesos, filesystem, etc) en Docker y Podman
- Mostrar ejemplos prácticos que permitan ver en clase la operatoria de ambas tecnologías.

Algunas páginas útiles:
https://docs.docker.com/
https://github.com/docker/labs
https://www.docker.com/what-docker
https://podman.io/
https://github.com/containers/podman

7. Pseudo filesystem proc de Linux

El directorio /proc en los sistemas GNU/Linux proporciona una interfaz para el kernel. Permite que las aplicaciones y los usuarios obtengan información y establezcan valores en el kernel utilizando la operación de E/S normal del sistema de archivos.
El sistema de archivos proc contiene archivos especiales, que poseen información del sistema en tiempo de ejecución (por ejemplo, memoria del sistema, dispositivos montados, configuración de hardware, etc.). Puede considerarse como un centro de control e información para el kernel del sistema. Muchas utilidades del sistema son simplemente llamadas a archivos en este directorio. Por ejemplo, el comando lsmod, es un "*cat /proc/modules*", mientras que lspci, un "*cat /proc/pci*". Al alterar los archivos ubicados en este directorio, es posible modificar los parámetros del kernel mientras el sistema esté en ejecución.
Realizar un informe y una exposición sobre la implementación del pseudo filesystem proc de Linux.
Puede considerar:

- Información sobre el hardware del sistema
- Interrupciones del sistema
- Módulos del sistema
- Información sobre la memoria del sistema
- Variables del kernel del sistema
- Variables de interés

Elegir un conjunto representativo de las variables para exponerlas en mayor detalle. Realizar pruebas en vivo de las variables elegidas.

Algunas páginas útiles:
https://en.wikipedia.org/wiki/Procfs
https://www.tldp.org/LDP/sag/html/proc-fs.html
http://man7.org/linux/man-pages/man5/proc.5.html
https://www.linux.com/news/discover-possibilities-proc-directory%20

8. Nginx

Nginx es un servidor web/proxy inverso ligero de alto rendimiento, además de servidor para protocolos de correo electrónico. Es una utilidad de software libre, licenciado bajo la Licencia BSD simplificada. Actualmente representa una de las mejores alternativas para la implementación de servidores web.
Realizar un informe y una presentación que contemple los siguientes puntos:

- Características técnicas de Nginx como servidor web
- Mecanismos de autenticación
- Posibilidades para el balanceo de carga
- Comparativas contra otras utilidades equivalentes
- Monitoreo de Nginx
- Utilidades que se sirven desde Nginx
- Ventajas y desventajas

Implementar un servidor web con Nginx. Debe considerar la puesta en funcionamiento de una página web, así como una utilidad a elección, provista desde Nginx.

Algunas páginas útiles:
https://www.nginx.com/
https://www.nginx.com/resources/wiki/
https://www.freecodecamp.org/news/an-introduction-to-nginx-for-developers-62179b6a458f/

9. LVM (Logical Volume Manager)

LVM es una implementación de un gestor de volúmenes lógicos para el kernel Linux. Incluye muchas de las características esperadas de un administrador de volúmenes, incluyendo redimensionado de grupos lógicos, redimensionado de volúmenes lógicos, instantáneas (snapshots) de volúmenes, entre otras.
Mediante la confección de un informe y una presentación del tema, investigar sobre LVM (Logical Volume Manager). Para ello, considerar los siguiente aspectos:

- Qué es LVM
- Funcionamiento y Arquitectura
- Snapshots de LVM: En qué consisten y para qué sirven
- Comandos para la manipulación de volúmenes lógicos
- Instalación y configuración de LVM
- Niveles de LVM:
    - Volúmenes Físicos (Physical Volumes)
    - Volúmenes Lógicos (Logical Volumes)
    - Grupos de volúmenes (Volume Groups)
    - Extensión Lógica (Logical Extents)
    - Extensión Física (Physical Extents)
- Ventajas y desventajas de uso

Construir ejemplos y/o casos que permitan mostrar en forma práctica los conceptos previamente expuestos. Utilizar casos reales y prácticos de uso de LVM.

Algunas páginas útiles:
https://es.wikipedia.org/wiki/Logical_Volume_Manager
http://tldp.org/HOWTO/LVM-HOWTO/
https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Logical_Volume_Manager_Administration/LVM_overview.html

Checkpoints de avances

A fin de monitorear el avance de los trabajos se establecerán checkpoints semanales, donde los integrantes de cada grupo comentarán y mostrarán el estado actual del TP. El propósito de los mismos es guiar al grupo durante la realización del trabajo, detectar y resolver problemas que impidan completar el mismo en el tiempo y forma esperados.

Se establece el siguiente cronograma de controles:

- **Jueves 02/06**
- **Jueves 09/06**
- **Jueves 16/06**

Exposiciones

Se fijan las siguientes fechas para la exposición de los trabajos ante el resto del curso:

- **Martes 21/06**
- **Jueves 23/06**

Observación: 48 Hs. previas al día asignado para la defensa del trabajo, el grupo deberá enviar el informe digital definitivo a la cátedra via mail, para una revisión previa a la exposición.

Forma de trabajo

El trabajo debe ser desarrollado en grupos de **no más de dos integrantes**.

Criterios de evaluación

El trabajo tendrá una nota grupal e individual; y se considerarán los siguientes puntos para su evaluación:

- Documento resultante de la investigación, en versión digital y subido al campus virtual.
- Presentación del tema en clase y material de soporte utilizado a tal fin (presentaciones,, ejemplos prácticos, etc).
- Casos prácticos usados en la investigación.
- Cumplimiento de los checkpoints propuestos por la cátedra.
- Trabajo en equipo.
- Exposición de cada uno de los integrantes.