**Trabajo Práctico N°1 – ISO.**

1. **GNU/LINUX:**
   1. Características de GNU/LINUX:

* Sistema operativo tipo Unix, pero LIBRE. Esto quiere decir que es un sistema que se comporta y opera de manera parecida a **Unix**, aunque no necesita ser certificado como una versión oficial de este. Unix es un tipo de S.O **multiusuario**, **multitarea** y **portátil**, conocido por su robustez y eficiencia.
* Gratuito y de libre distribución
* Posee diversas distribuciones y intérpretes de comando
* Es case sensitive y de código abierto, permite al usuario estudiarlo, personalizarlo, aprovecharse de la documentación y otras cosas.
  1. Sistemas Operativos en comparación a GNU/LINUX:
* Windows:
* Sistema operativo desarrollado por Microsoft, no es de tipo Unix, sino de MS-DOS
* Es de pago, con licencia. No es libre distribución
* Posee distintas versiones del sistema, no distribuciones
* No es de código abierto, por ende, no es accesible a su KERNEL para modificaciones del mismo
* MacOS:
* Es un sistema operativo tipo Unix, de pago. Es certificado como una versión oficial de este
* Es gratuito, pero no libre; viene incluido sin costo adicional en la compra de una Mac y se obtiene como una actualización de software gratuita para los equipos compatibles
* No posee distribuciones
* No es de código abierto
  1. Las siglas GNU vienen de = **G**NU **N**ot **U**nix (GNU No es Unix). Es un sistema operativo que permite a los usuarios acceder a su código para estudiarlo o realizar modificaciones de él.
  2. **GNU** es un proyecto iniciado por **Richard Stallman** con el fin de crear un Unix Libre (GNU). El mismo necesitó de un marco regulatorio conocido como **GPL** (General Public License de GNU).

Stallman creo la **Free Software Foundation** (FSF) para poder financiar el proyecto. El problema de GNU es que faltaba un componente principal, denominado **KERNEL.** Si bien se venía trabajando en un núcleo conocido como **TRIX**, el mismo fue abandonado dado su complejidad (Solamente corria en hardware muy costoso para la época). Cómo alternativa, Stallman, decidió adoptar como base el **Núcleo MACH** para crear **GNU Hurd**, que tampoco prosperó. Fue así cómo aparece **Linus Torvalds**, quien ya venia trabajando en un Núcleo (KERNEL) denominado **LINUX** el cuál se distribuiría bajo la licencia GPL. En el 92’, Torvalds y Stallman fusionan sus proyectos haciendo así GNU/Linux.

* 1. La multitarea es una función que poseen los sistemas operativos de realizar varias tareas al mismo tiempo por parte del usuario. Por ejemplo; Tener 2 ventanas abiertas, realizaron trabajos de impresión mientras el usuario escribe un texto, etc… GNU/Linux hace uso de la multitarea. En ello se destacan los tipos de S.O Unix.
  2. **POSIX** es una familia de estándares del IEEE que define un conjunto de interfaces de sistema, shells y utilidades comunes para garantizar la **compatibilidad** y **portabilidad** del software entre diferentes sistemas operativos. Esto significa que un programa escrito para cumplir con POSIX en una distribución GNU/Linux debería ejecutarse sin problemas en otra, así como en otros sistemas operativos basados en UNIX.

1. **Distribuciones de GNU/Linux:**
   1. Una distribución de GNU/Linux es una customización formada por una versión de **KERNEL** y determinados programas con sus configuraciones. Algunas de las distribuciones que podemos conocer son **DEBIAN, LINUX MINT, UBUNTU** o **FEDORA.**

**Ubuntu**, ideal para principiantes y usuarios en general; **Debian**, una base estable para muchas otras distros; **Linux** **Mint**, para una transición sencilla desde Windows; y **Fedora**, para usuarios que quieren software de vanguardia.

* 1. Una distribución de otra puede diferenciarse en base a su propósito, interfaz gráfica, utilidad, legibilidad y demás características. Como ejemplo, están aquellas que nombramos anteriormente.
  2. Debian es una distribución de GNU/Linux reconocida por varios usuarios, desarrolladores y entornos empresariales. Se valora a Debian por la estabilidad y fluidez de los procesos de actualización, como de los paquetes de toda la distro. Funciona en numerosas arquitecturas y dispositivos.

El proyecto **Debian** nace en Agosto de 1993, cuando **Ian Murdock** anuncia su creación en un mensaje a la lista de correo “**comp.os.linux.development”** . El nombre proviene de la combinación de su nombre y el de su entonces novia (esposa después), **Debra: Deb + Ian**. En los años 1994-1995 se publican **las primeras versiones** (0.9x) y se establecen los **Principios del Software Liibre de Debian**, donde se crea el **Contrato Social de Debian**, su base filosófica. De allí hasta la actualidad se han lanzado varias versiones de Debian con distintas características y soportes de arquitecturas. Al día de hoy su versión más reciente a lanzar es **Debian 13 “Trixie”**.

Tiene como objetivo crear un S.O libre, disponible para todos. Se refieren a “libre” como LIBERTAD del SOFTWARE. Esa es su filosofía.

1. **Estructura de GNU/Linux:**
   1. Los 3 componentes fundamentales de GNU/Linux son:

* **KERNEL**
* **HERRAMIENTAS DE GNU**
* **El Intérprete de comandos (Shell)**
  1. Estructura básica del sistema operativo de GNU/Linux:
* **NÚCLEO**: Conocido como **KERNEL**. Es el que ejecuta programas y gestiona dispositivos de hardware. Se encarga de tratar que el software y hardware trabajen juntos. Es **monolítico híbrido**, esto quiere decir que los drivers se ejecutan en modo privilegiado o modo kernel. Lo hace hibrido la capacidad de cargar y descargar funcionalidad a través de módulos
* **INTERPRETE** **DE** **COMANDOS**: Conocido como CLI (Command Line Interface), es el modo de comunicación entre un usuario y el S.O. Ejecuta programas a partir del ingreso de comandos.
* **SISTEMA DE ARCHIVOS**: Organiza la forma en que se almacenan los archivos en dispositivos de almacenamientos, el adoptado por GNU/Linux es el Extended (v2,v3,v4). Esta organizado según FHS (Filesystem Hierarchi Standard = Estándar de Jerarquía del Sistema de Archivos).
* **UTILIDADES:** Paquetes de software que permiten diferenciar las distribuciones. Poseen editores de texto, herramientas de networking, paquetes de oficina o interfaces gráficas.

1. **Kernel:**
   1. La funcion/es principal/es del kernel es gestionar los recursos de hardware y de software del sistema. Encargado de hacer que estos trabajen juntos.

Entre las funciones más importantes se encuentran la administración de memoria, la gestión de la CPU y E/S. Es decir, el corazón del S.O.

* 1. La versión actual es la **6.16.3**.

Definición del esquema de versionado del KERNEL en versiones anteriores a **2.4** y a partir de la **2.6**: Nomenclatura A.B.C.(D)

* **A:** Denota versión. Cambia con menor frecuencia. En 1994 (1.0), en 1996 (2.0), etc…
* **B:** Denota mayor revisión. Antes de la versión 2.6, los numeros impares indican desarrollo, los pares producción. Las versiones

impares eran consideradas más experimentales y menos estables, mientras que las versiones pares se consideraban más estables y adecuadas para su uso en sistemas de producción.

* **C:** Denota menor revisión. Solo cambia cuando hay nuevos drivers o características.
* **D:** Cambia cuando se corrige un error grave sin agregar nueva funcionalidad.

Esquema de Versionado Anterior a la 2.4 = Modelo "Par/Impar"). a partir de la versión 2.6 = Fin del Modelo Par/Impar.

* 1. Si, es posible tener más de 1 kernel GNU/Linux en nuestra máquina. Estos pueden iniciarse mediante un **gestor de arranque multiple** (**GRUB**), también conocido como un BOOTLOADER (Su finalidad es la de cargar la imagen del kernel -S.O- de alguna partición para su ejecución).
  2. El kernel, dentro del File System, se encuentra ubicado en **/boot**

1. **Intérprete de comandos (Shell):**
   1. Conocido como CLI (Command Line Interface), es el modo de comunicación entre un usuario y el S.O. Ejecuta programas a partir del ingreso de comandos.
   2. Su objetivo es brindar un modo de comunicación entre el usuario y el SO ejecutando programas a partir del ingreso de comandos.

* **Bourne Shell (sh);** Es el **estandar mínimo** Los scripts que empiezan con **#!/bin/sh** deben usar solo esta sintaxis básica para garantizar la máxima portabilidad en cualquier sistema Unix-like.
* **Korn Shell (ksh);** Fue una **mejora poderosa** que popularizó el shell scripting avanzado y la interactividad, dominando en entornos Unix comerciales.
* **Bourn Again Shell (bash);** Es el shell más popular y completo hoy en día. Es el estándar de facto en Linux y macOS, combinando lo mejor de sh y ksh y añadiendo muchas características propias. Es la mejor opción para uso interactivo y scripting moderno.
  1. Los comandos propios (internos) no tienen una ruta (path) física. Estan cargados en la memoria RAM del proceso del SHELL. Ejemplos: cd, echo, export, etc…

Los comandos externos (Ejecutables) se ubican (path) en directorios específicos.

* **Externos (Esencial):** /bin, /usr/bin
* **Externos (Admin):** /sbin, /usr/sbin
* **Externos (Usuario):** ~/.local/bin
* **Externos (Manual):** /usr/local/bin
  1. Considero que el SHELL no es parte del Kernel de GNU/Linux dado que Unix-Like tiene la arquitectura y filosofía de que cada componente debe hacer una sola cosa de manera independiente. El kernel y el shell tienen roles completamente distintos.

Además; El kernel opera en **modo kernel** (con acceso total al hardware), mientras que el shell se ejecuta en **modo usuario** (con restricciones). Si el shell fuera parte del kernel, cualquier vulnerabilidad en él (como inyección de comandos) comprometería directamente el núcleo del sistema, lo que sería un riesgo crítico de seguridad.

* 1. En GNU/Linux cada usuario tiene asignado un shell por defecto que se ejecuta cuando inicia sesión. Además, cada shell por defecto puede ser diferente (bash, zsh, fish, ksh, etc.).

El shell se define en el campo final de su entrada en el archivo /etc/passwd.

Un usuario puede cambiar su propio shell (usando chsh) siempre que el nuevo shell esté en **/etc/shells**. Solo el **root** puede cambiar el shell de otro usuario.

1. **El sistema de Archivos (File System) en Linux:**
   1. Un sistema de archivos organiza la forma en que se almacenan los archivos en dispositivos de almacenamientos, el adoptado por GNU/Linux es el Extended (v2,v3,v4).
   2. La **estructura básica** de los File System en GNU/Linux sigue el Estándar de **Jerarquía del Sistema de Archivos** (FHS - Filesystem Hierarchy Standard). El sistema de archivos se organiza como un único árbol que comienza desde el directorio raíz (/). Todos los demás directorios "cuelgan" de este.

Directorios más importantes:

* **/** = Tope de la estructura de directorios. Es como el C:\
* /**home** = Se almacenan archivos de USUARIOS (Mis Docs)
* /**var** = Información que varía de tamaño (Iogs, BD, spools)
* /**etc** = Archivos de configuración
* /**bin** = Archivos binarios y ejecutables
* /**dev** = Enlace a dispositivos
* /**usr** = Apps de usuarios.

Las siglas **FHS** hacen referencia a (Filesystem Hierarchi Standard = Estándar de Jerarquía del Sistema de Archivos).

* 1. Los sistemas de archivos **soportados** por GNU/Linux son:
* ext2, ext3, ext4 (Fourth Extended) 🡪 Estandar por defecto
* XFS
* Btrfs
* ZFS
* F2FS
* ReiserFS
* VFAT (FAT16/FAT32)
* exFAT
* NTFS
  1. Si, es posible dado que es soportado en GNU/Linux. Puede operar sobre ellos sin necesidad de un software adicional par FAT, sino con herramientas integradas en el kernel moderno o drivers para NTFS.

1. **Particiones:**
   1. Es una forma de dividir **lógicamente** el disco físico. Los tipos de particiones que existen son:

* **Partición Primaria**: División cruda del disco (Puede haber hasta 4 por disco). Se almacena información de la misma en el Master Boot Record (mbr)
* **Partición Extendida**: Sirve para contener unidades lógicas en su interior. Solo puede existir una partición de este tipo por disco. No se define un tipo de File System directamente sobre ella
* **Partición Lógica**: Ocupa la totalidad o parte de la partición extendida y se le define un tipo de File System. Las particiones de este tipo se conectan como una lista enlazada.

La **ventaja** de las particiones es que se considera una **buena práctica** al momento de separar los datos del usuario de las aplicaciones y/o sistemas operativos instalado. Se puede tener una partición de **restore** de todo el sistema. Poder **ubicar el KERNEL en una partición de solo lectura**, o una que ni siquiera se monta (No disponible para usuarios).

La **desventaja** de particionar demasiado un disco se nota al momento de gestionar sobre él. Se requiere decidir anticipadamente el tamaño de cada partición, dado que redimensionarlas después es **complejo y riesgoso**. Así mismo esto genera una **fragmentación de espacio**; El espacio libre se divide en muchos bloques pequeños imposibilitando almacenar archivos grandes, aunque el espacio total libre sea suficiente.

* 1. En GNU/Linux, las particiones se identifican mediante un esquema de nombres consistente basado en el tipo de controlador y la topología física.
* **SATA (Modernos):** Estos discos son tratados por el kernel como dispositivos SCSI, independientemente de su interfaz física. Utilizan la **Nomenclatura:** /dev/sdX#

🡪 SD: Indica un disco de tipo SCSI/SATA/USB.

🡪 X: Letra que identifica el disco.

-> #: El número que identifica la partición dentro del disco

* + Ejemplo:
    - **/dev/sda:** Primer disco SATA/SCSI/USB.
    - **/dev/sdb3**: Tercera partición del segundo disco.
* **IDE (Obsoletos, pero relevantes):** Aunque ya no son comunes, se identifican de forma diferente.
* Utilizan la **Nomenclatura:** /dev/hdX#

-> hd: Indica un disco IDE/PATA

-> X: Letra que identifica el disco y el canal IDE

-> #: Número de la partición

* + Ejemplo:
    - **/dev/hda:** Primer disco IDE (primary master).
    - **/dev/hdc2:** Segunda partición del tercer disco IDE (secondary master).
  1. Como mínimo se necesita 1 partición:
* **Tipo**: Primaria o Extendida/Logical
* **Identificación**: /dev/sda1 (Ejemplo)
* **FS**: ext4 (o XFS/Btrfs)
* **Punto de montaje**: **/** (Raíz)

De todas maneras, se **recomiendan** básicamente 2 particiones:

**Partición SWAP:**

* **Tipo**: Primaria o Extendida/Logical
* **Identificación**: /dev/sda2 (Ejemplo)
* **FS**: swap
* **Punto de montaje**: No se monta,s e activa con **swapon**

**Partición Raíz (/):**

* **Tipo**: Primaria o Extendida/Logical
* **Identificación**: /dev/sda1 (Ejemplo)
* **FS**: ext4
* **Punto de montaje**: **/** (Raíz)
  1. –
  2. Para crear particiones, se utiliza un software llamado **particionador**, que puede ser **destructivo**, permitiendo crear y eliminar particiones, o **no** **destructivo**, permitiendo crear, eliminar y modificar particiones existentes. Las herramientas de particionamiento incluyen **fdisk** para particionamiento destructivo y **gparted** o **ps** para no destructivo. Muchas distribuciones de sistema operativo tienen una interfaz gráfica de instalación que permite crear particiones durante el proceso de instalación.