



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorio de Computacion Salas A y B

Profesor(a):

Asignatura:

Grupo:

No de practica(s):

Integrante(s):

No de lista o brigada:

Semestre:

Fecha de entrega:

Observaciones:

Calificacion:

Cuestionario previo Práctica 2

1. ¿Qué es un sistema operativo?

Un sistema operativo (SO) es un software que actúa como intermediario entre los usuarios de una computadora y el hardware de la misma. Su función principal es gestionar los recursos del hardware y proporcionar un entorno en el que los programas de aplicación puedan ejecutarse.

Algunas de las tareas específicas que realiza un sistema operativo incluyen:

1. Gestión de procesos: Controla la creación, ejecución, y finalización de procesos, y gestiona la asignación de recursos entre ellos.
2. Gestión de memoria: Administra la memoria principal (RAM) del sistema, asignando espacio de memoria a los procesos cuando lo necesitan y liberándolo cuando ya no es necesario.
3. Gestión de almacenamiento: Controla la lectura y escritura de datos en dispositivos de almacenamiento, como discos duros, SSDs, y unidades extraíbles.
4. Gestión de dispositivos: Facilita la comunicación entre el software y el hardware, como impresoras, teclados, y ratones, asegurándose de que los dispositivos funcionen correctamente.
5. Interfaz de usuario: Proporciona una interfaz para que los usuarios interactúen con la computadora, ya sea a través de una interfaz gráfica (GUI) o una interfaz de línea de comandos (CLI).

Ejemplos populares de sistemas operativos son Windows, macOS, Linux, y Android.

2. ¿Qué es la terminal?

La terminal es una interfaz de línea de comandos (CLI) que permite a los usuarios interactuar con el sistema operativo a través de comandos textuales en lugar de una interfaz gráfica (GUI). Es una herramienta poderosa en programación y administración de sistemas, ya que permite realizar tareas de manera más directa y a menudo más rápida que utilizando una interfaz gráfica.

Características

1. Ejecución de comandos: Se pueden escribir comandos que el sistema operativo ejecutará, como navegar por directorios, copiar archivos, instalar software, o compilar código.
2. Automatización: Se pueden escribir scripts (conjuntos de comandos) para automatizar tareas repetitivas, como la instalación de dependencias, la compilación de proyectos, o la ejecución de pruebas.
3. Acceso a herramientas de desarrollo: Muchos lenguajes de programación y entornos de desarrollo ofrecen herramientas de línea de comandos que se ejecutan en la terminal, como git para control de versiones, npm para gestionar paquetes en JavaScript, o pip en Python.

3. ¿Cómo se crea un archivo?

Crear un archivo en programación depende del lenguaje que estés utilizando. Sin embargo, el proceso generalmente implica abrir o crear el archivo, escribir en él, y luego cerrarlo. A continuación, te mostraré cómo hacerlo en algunos lenguajes de programación populares:

En Python, puedes crear un archivo usando la función `open()` y especificando el modo de apertura. Si el archivo no existe, Python lo creará.

En Java, puedes crear un archivo usando la clase `FileWriter`.

En C, usas la función `fopen()` para crear y abrir un archivo.

En un entorno de Node.js, puedes usar el módulo `fs` para trabajar con archivos.

En un script de Bash, puedes crear un archivo usando la redirección `>` o `touch`.

4. ¿Cómo se mueve un archivo?

Mover un archivo implica cambiar su ubicación en el sistema de archivos, y esto se puede hacer de diferentes maneras dependiendo del lenguaje de programación o entorno que estés utilizando. Aquí te muestro cómo hacerlo en algunos casos comunes:

En sistemas Unix (Linux, macOS), puedes usar el comando `mv` para mover archivos.

En Python, puedes mover un archivo usando la función `shutil.move()` del módulo `shutil`.

En Java, puedes usar la clase `Files` del paquete `java.nio.file` para mover un archivo.

5. ¿Cómo se copia un archivo?

Copiar un archivo significa crear una copia del archivo en una ubicación diferente o con un nombre diferente.

En sistemas Unix (Linux, macOS), se puede usar el comando `cp` para copiar archivos.

En Python, puedes usar la función `shutil.copy()` del módulo `shutil` para copiar archivos.

En Java, puedes usar la clase `Files` del paquete `java.nio.file` para copiar un archivo.

DESARROLLO DE PRÁCTICA

```
estudiante@Camerun33 ~ % nano
UW PICO 5.09                               New Buffer                               Modified
Me visualizo terminando mi carrera, con mi título y con una vida llena de
experiencias de todo tipo.

estudiante@Camerun33 ~ % mkdir Algebra
estudiante@Camerun33 ~ % mkdir Programacion
estudiante@Camerun33 ~ % mkdir Redaccion
estudiante@Camerun33 ~ % mkdir Calculo
estudiante@Camerun33 ~ % mkdir IngenieriaIndustrial
estudiante@Camerun33 ~ % ls
Algebra          Jesus            Pictures
Calculo          Jesús García Buffer  Programacion
Desktop          Jesús.txt         Public
Documents        Library          Redaccion
Downloads        Movies           fp
IngenieriaIndustrial Music
estudiante@Camerun33 ~ %
```

```

[estudiante@Camerun33 ~ % ls
Desktop          Downloads        Library          Music            Public
Documents        Jesús.txt       Movies           Pictures         fp
[estudiante@Camerun33 ~ % nano
[estudiante@Camerun33 ~ % nano Jesus
[estudiante@Camerun33 ~ % mkdir Algebra
[estudiante@Camerun33 ~ % mkdir Programacion
[estudiante@Camerun33 ~ % mkdir Redaccion
[estudiante@Camerun33 ~ % mkdir Calculo
[estudiante@Camerun33 ~ % mkdir IngenieriaIndustrial
[estudiante@Camerun33 ~ % ls
Algebra          Jesus           Pictures
Calculo          Jesús García Buffer Programacion
Desktop          Jesús.txt       Public
Documents        Library         Redaccion
Downloads        Movies          fp
IngenieriaIndustrial Music
[estudiante@Camerun33 ~ % cd programacion
[estudiante@Camerun33 programacion % ls
[estudiante@Camerun33 programacion % nano Algoritmo.txt
[estudiante@Camerun33 programacion % ls
Algoritmo.txt
[estudiante@Camerun33 programacion % █

```

```

├── Programacion
│   ├── Algoritmo.txt
│   ├── Codigo.txt
│   ├── Condicional.txt
│   ├── Lenguaje.txt
│   └── Variable.txt

```

```

├── Algebra
│   ├── Coseno.txt
│   ├── Monomio.txt
│   ├── Seno.txt
│   ├── Tangente.txt
│   └── Trigonometria.txt
├── Calculo
│   ├── Conicas.txt
│   ├── Derivada.txt
│   ├── Integral.txt
│   ├── Limite.txt
│   └── serie.txt

```

```

[estudiante@Camerun33 ~ % ls /
Applications  Library      System      Users      Volumes    bin      cores      dev      etc      home      opt      private    sbin      tmp      usr      var
[estudiante@Camerun33 ~ % █

```

```

[estudiante@Camerun33 ~ % touch Jesús.txt
[estudiante@Camerun33 ~ % ls
Desktop          Downloads        Library          Music            Public
Documents        Jesús.txt       Movies           Pictures         fp
[estudiante@Camerun33 ~ % █

```

```

[estudiante@Camerun33 ~ %
estudiante@Camerun33 ~ % ls -l
total 0
drwx-----+ 6 estudiante staff 192 Aug 19 17:12 Desktop
drwx-----+ 5 estudiante staff 160 Aug 19 09:50 Documents
drwx-----+ 3 estudiante staff 96 Aug 19 07:05 Downloads
drwx-----@ 78 estudiante staff 2496 Aug 19 11:18 Library
drwx----- 3 estudiante staff 96 Aug 19 07:05 Movies
drwx-----+ 3 estudiante staff 96 Aug 19 07:05 Music
drwx-----+ 4 estudiante staff 128 Aug 19 08:15 Pictures
drwxr-xr-x+ 4 estudiante staff 128 Aug 19 07:05 Public
[drwxr-xr-x 3 estudiante staff 96 Aug 19 13:16 fp
[estudiante@Camerun33 ~ %
estudiante@Camerun33 ~ % ls /
Applications Library System Users Volumes
bin cores dev etc home
opt private sbin tmp usr
var
[estudiante@Camerun33 ~ % touch Jesús.txt
[estudiante@Camerun33 ~ % ls
Desktop Downloads Library Music Public
Documents Jesús.txt Movies Pictures fp
estudiante@Camerun33 ~ % █

```

```

├─ Downloads
├─ IngenieriaIndustrial
│   └─ Industria.txt
│       └─ Planeacion.txt
├─ Jesus
├─ Jesús García Buffer
├─ Jesús.txt
└─ Library

```