**Variables y parámetros**

* **Variables del sistema:**
  + printenv: Visualiza todas las variables del sistema.
  + set: Sin parámetros, muestra la lista de variables en curso. Con set variable=valor, se puede cambiar el valor de una variable.
  + readonly variable: Impide que una variable se modifique.
  + unset variable: Elimina una variable.
* **Variables de entorno:**
  + export variable: Permite hacer una variable de sesión disponible para subshells.
  + ~/.bash\_profile: Configuración específica de un usuario individual.
  + /etc/profile: Configuración global que se ejecuta al iniciar sesión en bash.
  + PATH: Variable que define los directorios donde se buscan los ejecutables.
* **Parámetros de posición:**
  + set valor1 valor2 valor3: Asigna valores a los parámetros de posición. Ejemplo: $1 es valor1, $2 es valor2, etc.
  + $0: Nombre del script.
  + $#: Número de argumentos pasados al script.
  + $\*: Lista de todos los argumentos pasados al script.
* **Otros parámetros:**
  + $$: PID del proceso en ejecución.
  + $variable: Accede al contenido de la variable.

**Historial y configuración**

* ~/.bash\_history: Archivo que almacena el historial de comandos ejecutados.
* ~/.bashrc: Archivo de configuración que se ejecuta cada vez que se abre una nueva shell.

**Comandos útiles**

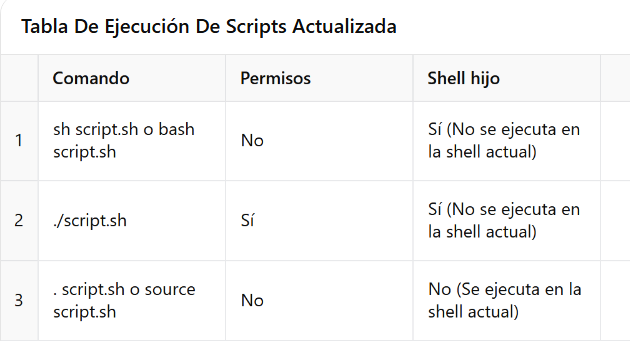
* echo -n: Imprime un mensaje sin salto de línea.
* echo -e "cadena\n": Imprime una cadena con un salto de línea.
* uname: Muestra el nombre del sistema operativo.
* umask: Establece la máscara de creación de archivos.
* ps: Muestra los procesos en ejecución.
* comando &: Ejecuta un comando en segundo plano.
* ln archivo1.txt enlace\_duro.txt: Crea un enlace duro.
* ln -s archivo enlace\_simbólico: Crea un enlace simbólico.
* let variable=operación: Realiza operaciones aritméticas y asigna el resultado a una variable.

**Expresiones y operadores**

* [ cadena1 = cadena2 ]: Verifica si dos cadenas son iguales. (Nota: los espacios dentro de los corchetes son obligatorios).
* test cadena1 = cadena2: Equivalente a la expresión anterior.
* "cadena": Permite entrecomillar caracteres especiales.
* 'cadena': Similar a las comillas dobles, pero no interpreta variables.
* \caracter: Escapa el significado especial de un carácter.

**Ejecutables y procesos**

* taskmgr.exe: Administrador de tareas (Windows).
* services.msc: Administración de servicios (Windows).
* taskschd.msc: Programador de tareas (Windows).
* msconfig.exe: Configuración del sistema (Windows).
* eventvwr.msc: Visor de eventos (Windows).
* DISPATCHER: Planificador de procesos. (Procesos)
* Middleware: Gestiona la distribución de datos y tareas entre servidores. (Granja)
* THROUGHPUT: Rendimiento del número de trabajos ejecutados por unidad de tiempo. (Procesos)
* Proceso: Ejecución activa de un programa.
* Servicio: Proceso en segundo plano.
* Programa: Proceso en ejecución.

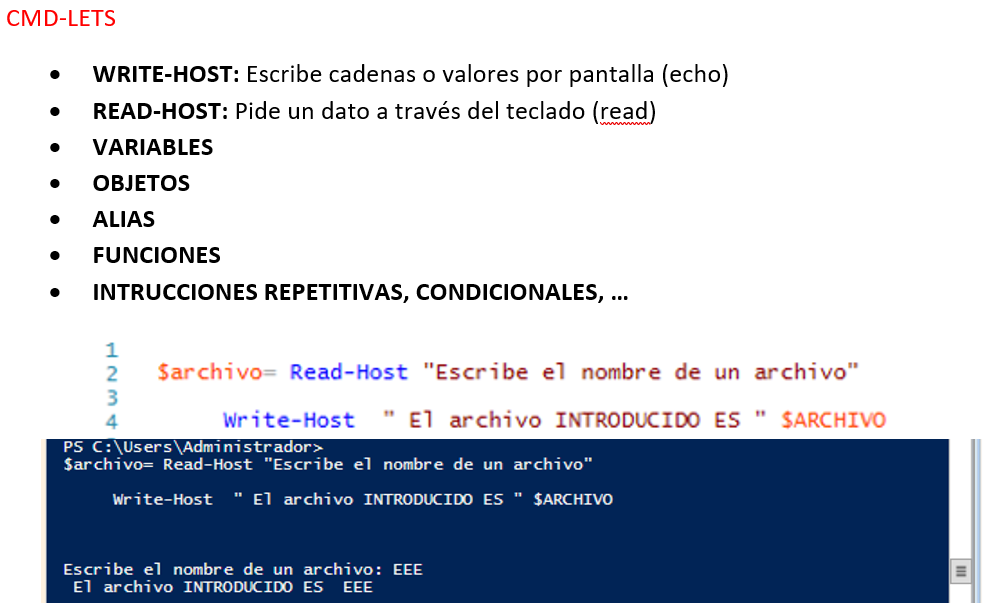


**4. echo -n "Linea 1"**

* Este comando genera la cadena "Linea 1" pero no genera un salto de linea al final.
* Si lo ejecutas en tu terminal, el texto "Linea 1" aparecerá, y el cursor quedara al final de la linea, sin saltar a la siguiente.

**3. echo -e "Linea 1\nLinea 2"**

* Este comando utiliza la secuencia de escape \n para crear un salto de línea dentro de la cadena de texto.
* La salida será:
  + Linea 1
  + Linea 2



1. **$archivo = Read-Host "Escribe el nombre de un archivo"**: (read -p)
   * Read-Host es un comando en PowerShell que se utiliza para solicitar información al usuario a través del teclado.
   * "Escribe el nombre de un archivo" es el mensaje que se muestra al usuario, pidiéndole que ingrese el nombre de un archivo.
   * El valor que el usuario ingresa se almacena en una variable llamada $archivo. En PowerShell, los nombres de variables comienzan con el símbolo $.
2. **Write-Host "El archivo INTRODUCIDO ES $ARCHIVO"**:
   * Write-Host es un comando en PowerShell que se utiliza para mostrar texto en la consola. Es similar al comando echo en otros shells como Bash.
   * "El archivo INTRODUCIDO ES $ARCHIVO" es el mensaje que se muestra al usuario.

CMD:

dsadd group "CN=ImpresorasASIR,CN=Users,DC=valle,DC=local" -secgrp yes -scope g

for /l %i in (1,1,30) do dsmod group "CN=ImpresorasASIR,CN=Users,DC=valle,DC=local" -addmbr "CN=asir%i,CN=Users,DC=valle,DC=local"

Abre "Usuarios y equipos de Active Directory" en la máquina virtual VALLE1.

**Configuración de Impresión Virtual**

1. **Instalación del Rol de Servicios de Impresión y Documentos:**
   * Abre el "Administrador del servidor" en la máquina virtual VALLE1.
   * Haz clic en "Agregar roles y características".
   * Selecciona "Instalación basada en características o en roles" y haz clic en "Siguiente".
   * Selecciona el servidor (VALLE1) y haz clic en "Siguiente".
   * Selecciona el rol "Servicios de impresión y documentos" y haz clic en "Agregar características".
   * Haz clic en "Siguiente" hasta llegar a la página de confirmación y haz clic en "Instalar".
2. **Creación de una Impresora Virtual:**
   * Abre el "Administrador de impresión" en la máquina virtual VALLE1.
   * Haz clic derecho en "Impresoras" y selecciona "Agregar impresora".
   * Introduce una dirección IP ficticia (por ejemplo, 192.168.10.100) y un nombre de puerto (por ejemplo, "ImpresoraVirtual").

**Configuración de Permisos de Seguridad:**

* Abre el "Administrador de impresión" en la máquina virtual VALLE1.
* Haz clic derecho en la impresora virtual creada y selecciona "Propiedades".
* Ve a la pestaña "Seguridad".
* Elimina el grupo "Todos" de la lista de usuarios y grupos.
* Agrega el grupo

Y compartir (tres opciones)

* Compartir esta impresora
* Presentar trabajos de impresión en equipos cliente
* Mostrar lista en el directorio