Apuntes de Lectura: Pro Bash Programming

Resumen Capítulo 1: Comandos y Conceptos Básicos

A continuación se pueden ver los **comandos** principales que se pueden utilizar en la terminal:

- **pwd:** Imprime el nombre del directorio en el que me encuentro.
- **cd:** Cambia del directorio actual al que se introduce como parámetro.

  ```
  cd destino
  ```

- **echo:** Imprime el argumento que se introduzca como parámetro.

  ```
  echo "mensaje"
  ```

- **mkdir:** Crea un nuevo directorio con nombre introducido como parámetro.

  ```
  mkdir NombreDirectorio
  ```

- **chmod:** Permite modificar los permisos que posee u archivo.

  ```
  chmod Permisos
  ```

- **printf:** Permite imprimir la cadena de caracteres que se de por parámetro.

  ```
  printf "Mensaje"
  ```

A continuación se enuncian los principales **conceptos** útiles en programación Bash:

- **Script:** Archivo que contiene comandos que son ejecutados por la terminal.
- **Comentarios:** Texto comenzado con numeral (#) el cual no es leído como código.

Resumen Capítulo 2: Parámetros y Salidas

Existen varios tipos de parámetros que se clasifican como se muestra a continuación:

- **De Posición:** Denotan, numéricamente, la pisición de un argumento, de la forma `$1` , `$2` , etc...
- **Variables:** Parámetros que guardan un valor numérico, de carcateres, etc...

A continuación se muestran comandos que permiten modificar la presentación de los argumentos a imprimir:

Comando	Efecto
`\a`	Alerta
`\b`	Backspace
`\e`	Escape Character
`\t`	TAB Horizontal
`\v`	TAB Vertical
`\\`	Backslash
`\nnn`	Caracter dado por código de 1 a 3 caracteres
`\xHH`	Caracter dado por código de 1 a 2 caracteres

A continuación se muestran los comandos que se pueden utilizar para especificar el formato de los argumentos dados como entrada al comando `printf` :

Comando	Efecto	Ejemplo
%s	Imprime lo que diga el argumento	`printf "%s\n" Mensaje`
%d	Indica que son argumentos enteros	`printf "%d\n" 25 46 58`
%f	Indica que son fraccionarios	`printf "%f\n" 23.6 7.8`
%e	Notación Científica	`printf "%e\n"`

A continuación se muestran los comandos que permiten administrar el espaciado en la línea de *output*:

Comando	Efecto	Ejemplo
`printf "%ns %-ms" Argumentos`	Organiza los alineamientos para correr cada argumento n y m espacios hacia la derecha si es positivo o izquierda si es negativo	`printf "%8s %-10:s" Mensaje` Nótese que el `:` será un caracter que estará fijo en esa posición.
`printf "%nd"`	Organiza la precisión de los decimales incluída.	`printf "%n.md" Argumento`

Ahora, en lugar de editar la terminal, es posible introducir el *output* en una variable, como se muestra a continuación:

`printf -v VARIABLE "especificación" Argumento`

Ahora, se verá como se realizan los métodos de **redirección** en Bash.

- **Redirección con > :** Este método permite dos grandes posibilidades. Crear un archivo al cual redireccionar el contenido, si es que no existía antes, o, en caso de que sí existiese, lo vacía e inserta el contenido deseado.

  `"mensaje/archivo/carpeta" > "mensaje/archivo/carpeta destino"`

- **Redirección con >> :** Este método es bastate similar al uso de `>` , sin embargo este no vacía el archivo, carpeta destino.

  `mensaje/archivo/carpeta <<`

Es posible también leer el contenido de una variable, esto se realiza con el comando `read` , el cual recibe como parámetro el nombre de la variable. Por ejemplo:

```
variable1 = Hola
read variable1
```

Este código debería arrojar como resultado el "Hola" contenido por la variable leída. También se pueden leer

varias variables:

```
variable1 = Hola
variable 2 = Amigos
read variable1 variable2
```

Este código me permite leer los contenidos "Hola" y "Amigos".

Pipelines

En caso de necesitar no solo imprimir contenido sino utilizarlo como entrada para **otro** comando, se utiliza `|`.

`printf "%s\n" Contenido Para Pasar | Comando2`

Por ejemplo, podría querer imprimir en consola un contenido y a la vez añadirlo a otro archivo.

```
printf "%s/n" 1 2 3 4 | tee ArchivoNuevo
```

Con el comando `tee` se imprime el contenido y se envía al archivo nuevo.