

Bourne Again Shell

Parte VI: Funciones

Dr. Mario M. Berón

Universidad Nacional de San Luis

Funciones Básicas de Script

Creación de una Función

```
function nombre {  
  comandos  
}
```

- nombre: Define el nombre asignado a la función.
- comandos: Son uno o más comandos que forman parte de la función. Dichos comandos se ejecutan cada vez que la función es invocada.

Funciones Básicas de Script

Creación de una Función

```
nombre() {  
    comandos  
}
```

Arriba se presenta un formato alternativo los paréntesis indican que lo que se está definiendo es una función.

Funciones Básicas de Script

Uso de Funciones - Invocación

- La invocación de una función se lleva a cabo colocando el nombre de una función en una línea del script como con cualquier otro comando.
- Cuando el intérprete encuentra la invocación de una función transfiere el control de la ejecución al cuerpo de la misma y ejecuta los comandos.
- Cuando el intérprete termina de ejecutar los comandos de la función retorna el control de la ejecución al próximo comando después de la invocación.

Funciones Básicas de Script

Nota

- Las definiciones de función pueden aparecer en cualquier lugar del script. No obstante, se debe tener cuidado de usar la función después que esta ha sido definida.
- Los nombres de las funciones son ÚNICOS. Si Ud. define una función con un nombre que ya ha sido utilizado por otra función. La segunda función sobre escribe la primera.

Retorno de un Valor

El Estado de Salida por Defecto

- El estado de salida retornado por una función está dado por el estado de salida del último comando que se ejecutó.
- Se puede utilizar el comando *return* para devolver un estado de salida específico. Cuando se utiliza esta técnica se debe tener en cuenta que:
 - El rango del estado de salida está comprendido entre 0 y 255.
 - Para que la función retorne cualquier otro valor y no sea tomado como un error se debe indicar el resultado con el comando *echo* y asignar a una variable el valor retornado por una función.

Uso de Variables en la Funciones

Parámetros

- Los parámetros de una función son identificados por variables especiales denotadas como: \$0,\$1,\$2,\$3,\$4,\$5,\$6... donde \$0 es el nombre de la función, \$1 es el primer parámetro, \$2 es el segundo parámetro y así siguiendo.
- Los parámetros de una función no son los mismos que los parámetros utilizados en la línea de comando.

Uso de Variables en la Funciones

Variables Globales

Son variables que son válidas en cualquier sitio del script.

- Si se define una variable en la sección principal del script, su valor se puede recuperar dentro de una función.
- Si una variable se define dentro de una función como global, su valor se puede recuperar dentro de la parte principal del script.
- Por defecto las variables definidas en el script son globales.

Uso de Variables en la Funciones

Variables Locales

Cualquier variable que la función utilice internamente puede ser declarada como *local*.

Ejemplo

```
local temp
```

```
local temp = $[ $value + 5 ]
```

Cualquier variable que la función utilice internamente puede ser declarada como *local*.

Creación de una Librería

Pasos

- 1 Crear un archivo con las funciones que se desean formen parte de la librería.
- 2 Incluir el archivo creado en el ítem previo en los scripts que necesitan usar las funciones definidas dentro de ese archivo.
 - Esta tarea se realiza usando el comando *source*. Este comando no crea una nueva instancia del intérprete de comando. Esta característica permite que las funciones de la librería puedan ser invocadas sin problemas dentro del script.
 - Un alias del comando *source* es *."*. Por lo tanto *./funciones* dentro de un script incorpora las funciones definidas en *funciones* dentro del ambiente del script.

Uso de Funciones en la Línea de Comandos

Creación de Funciones en la Línea de Comandos

- Definir toda la función en la línea de comandos.

```
$ function doubleit { read -p "Enter value: " value;  
echo [$ value * 2 ]; }  
$ doubleit  
Enter value: 20  
40  
$
```

Es importante notar que se debe colocar ";" al final de cada comando.

Uso de Funciones en la Línea de Comandos

Creación de Funciones en la Línea de Comandos

- Usar múltiples líneas para definir la función.

```
$ function mult {  
> echo $[ $1 * $2 ]  
> }
```

Uso de Funciones en la Línea de Comandos

Definición de Funciones en el Archivo `.bashrc`

- Definir las funciones directamente en el archivo `.bashrc`.
- Cargar las funciones definidas en una librería usando el comando *source*.