

## **Desde Linux**

Menú

Buscar...

Buscar

---

- [Inicio](#)
- [GNU/Linux](#)
- [Aplicaciones](#)
- [Apariencia](#)
- [Distribuciones](#)
- [Noticias](#)
- [Tutoriales](#)
- [Login / Registro](#)

- o [Login](#)
- o [Registro](#)



## eToro™ - Award-Winning Broker

Online Stock Trading Platform

Consult & Copy Leading Traders in Real Time. Invest in Stocks Today with as Little as \$200

content.etoro.com

[ABRIR](#)

[Inicio](#) » [GNU/Linux](#) » **Como extraer parámetros desde el Terminal aplicando Shell Scripting**

## Como extraer parámetros desde el Terminal aplicando Shell Scripting

6 de enero de 2016 a las 02:14.

[7 Comentarios](#)

BÚSQUEDAS PATROCINADAS

1 script
35 Printing

3d arquitectura
3d CAD files

Comparte
Twitea
Envía
Share on Google+
Pinea
Email
Print

El **Shell Scripting**, se refiere a la **ejecución de órdenes complejas** sobre el **Terminal (Consola) de GNU/Linux**, es muy útil para automatizar actividades rutinarias e importantes dentro de nuestro **Sistema Operativo GNU/Linux**, lo que permite optimizar nuestros **Recursos y Tiempo**, es decir, con lo aquí contenido exploraremos como desde el terminal podemos ejecutar órdenes de comando manuales que luego nos permitan **programar/automatizar** actividades sobre un equipo ahorrando **Horas/Labor** de ejecución manual o presencial, implementando dichas órdenes en un Script de Bash Shell o compatible y explicado de una manera, práctica y sencilla.

## Shell Scripting

Abarcaremos como en **una sola orden de comando** podemos **extraer y mostrar valores / información del Sistema Operativo / Hardware**, que luego podemos implementar dentro de un **Script de Bash Shell** para automatizar una tarea específica. Concentrándonos en las **Mejores Prácticas (Best Practices)** necesarias para obtener un excelente diseño de un Script de la manera **más eficiente y práctica**.

### Obtener el nombre del primer usuario creado en el sistema:

BÚSQUEDAS PATROCINADAS

3d CAD files
3d CAT

3d Character
3d Home

```
USER_1000=$(cat /etc/passwd | grep 1000 | cut -d: -f1) ; echo $USER_1000
```

```
USER_1001=$(cat /etc/passwd | grep 1001 | cut -d: -f1) ; echo $USER_1001
```

**Obtener la ruta del /home del primer usuario creado en el sistema:**

```
USER_1000=$(cat /etc/passwd | grep 1000 | cut -d: -f1) ; HOME_USER_1000=/home/$USER_1000 ; echo $HOME_USER_1000
```

BÚSQUEDAS PATROCINADAS

3d Home

3d Model

3d print

3d Program

```
USER_1001=$(cat /etc/passwd | grep 1001 | cut -d: -f1) ; HOME_USER_1001=/home/$USER_1001 ; echo $HOME_USER_1001
```

**Comprobar la Fecha actual del Equipo:**

```
FECHA_ACTUAL=$(date +"%d %b %y") ; echo $FECHA_ACTUAL
```

**Comprobar la Hora actual del Equipo:**

```
HORA_ACTUAL=$(date +"%H:%M") ; echo $HORA_ACTUAL
```

**Comprobar si el Host tiene conexión a Internet:**

```
if ping -c 1 8.8.8.8 &> /dev/null; then CONEXION_INTERNET=Habilitado; else CONEXION_INTERNET=Deshabilitado; fi ; echo $CONEXION_INTERNET
```

```
TEST_PING=$(ping 192.168.3.249 -c 5 | grep packet | awk '{print $6}' | cut -f1 -d%) ; echo $TEST_PING % de Perdida de paquetes
TEST_LATENCIA=$(ping 8.8.8.8 -c 5 | grep packet | awk '{print $10}' | cut -f1 -d%) ; echo $TEST_LATENCIA de Latencia del Enlace
```

**Comprobar el tipo de Sistema Operativo:**

```
SISTEMA_OPERATIVO=$(uname -o) ; echo $SISTEMA_OPERATIVO
```

**Comprobar el nombre, versión y subversión del Sistema Operativo:**

```
NOMBRE_SISTEMA=$(cat /etc/os-release | grep NAME | grep -v "VERSION" | sed -n '2p' | cut -f2 -d\") ; echo $NOMBRE_SISTEMA
VERSION_SISTEMA=$(cat /etc/os-release | grep VERSION= | sed -n '1p' | sed 's/VERSION=//' | sed 's//g') ; echo $VERSION_SISTEMA
SUBVERSION_SISTEMA=$(lsb_release -d | awk '{print $4}') ; echo $SUBVERSION_SISTEMA
```

**Comprobar la arquitectura del Sistema Operativo:**

```
ARQUITECTURA=$(uname -m) ; echo $ARQUITECTURA
```

```
ARQUITECTURA=$(uname -m) ; if [[ "$ARQUITECTURA" = "x86" ]]; then ARQ_SISTEMA=32; else ARQ_SISTEMA=64; fi ; echo $ARQ_SISTEMA
```

**Comprobar la versión del kernel del Sistema Operativo:**

```
VERSION_KERNEL=$(uname -r) ; echo $VERSION_KERNEL
```

**Comprobar el nombre del Host:**

```
NOMBRE_HOST=$(cat /etc/hostname) ; echo $NOMBRE_HOST
```

**Comprobar IP Interna y Externa (Principal de Salida a Internet):**

```
IP_INTERNA=$(hostname -I) ; echo $IP_INTERNA
IP_EXTERNA=$(curl -s ipecho.net/plain;echo) ; echo $IP_EXTERNA
```

**Comprobar Valores (IP / MAC) de las Interfaces de Red:**

```
IP_ETH0=$(ifconfig eth0 | grep inet | grep -v inet6 | cut -d ":" -f 2 | cut -d " " -f 1) ; echo $IP_ETH0
MAC_ETH0=$(ifconfig eth0 | sed -n '1p' | awk '{print $5}') ; echo $MAC_ETH0
IP_WLAN0=$(ifconfig wlan0 | grep inet | grep -v inet6 | cut -d ":" -f 2 | cut -d " " -f 1) ; echo $IP_WLAN0
MAC_WLAN0=$(ifconfig wlan0 | sed -n '1p' | awk '{print $5}') ; echo $MAC_WLAN0
```

**Comprobar el Proxy / Gateway del Sistema Operativo:**

```
PROXY_GATEWAY=$(route -n | sed -n '3p' | awk '{print $2}') ; echo $PROXY_GATEWAY
```

**Comprobar el Nombre de Dominio de Red del Host:**

```
DOMINIO=$(cat /etc/resolv.conf | sed '2 d' | grep search | sed -n '1p' | awk '{print $2}') ; echo $DOMINIO
```

**Comprobar la Dirección de Red (IP) del Servidor DNS del Host:**

```
IP_SERVIDOR_DNS=$(cat /etc/resolv.conf | sed '1 d' | awk '{print $2}') ; echo $IP_SERVIDOR_DNS
```

**Comprobar usuarios conectados al Host:**

```
who>/tmp/who ; echo -e "Usuarios conectados al Host :" && echo "" && echo "Usuarios Puertos      Fecha      Hora  Pantalla" ; echo
"*****" && cat /tmp/who

USUARIOS_CONECTADOS=$(who | awk '{print $1}') ; echo $USUARIOS_CONECTADOS

USER_ONLINE1=$(who | awk '{print $1}') ; echo $USER_ONLINE1 | wc -w

USER_ONLINE2=$(top -n 1 -b | grep "load average:" | awk '{print $6}') ; echo $USER_ONLINE2
```

**Comprobar Carpeta de Usuario con más data (# de Archivos / Tamaño en Bytes) en el Sistema Operativo:**

```
CARPETA_USUARIO1=$(ls -l /home | sed '1 d' | sort -k2 | sed q | awk '{print $9}') ; echo $CARPETA_USUARIO1
DATA_USUARIO1=$(du -sh /home/* | sort -r | sed q | awk '{print $1}') ; echo $DATA_USUARIO1
```

**Comprobar Tamaño en Bytes de Carpeta de Superusuario u otra Carpeta del Sistema Operativo:**

```
DATA_ROOT=$(du -sh /root | awk '{print $1}') ; echo $DATA_ROOT
DATA_CARPETA1=$(du -sh /var | awk '{print $1}') ; echo $DATA_CARPETA1
```

**Comprobar Carpetas de usuarios creados (Nombres y Números):**

```
NOMBRES_CARPETAS=$(ls -l /home | sed '1 d' | awk '{print $9}') ; echo $NOMBRES_CARPETAS
```

```
NUMERO_CARPETAS=$(ls -l /home | sed '1 d' | awk '{print $9}') ; echo $NUMERO_CARPETAS | wc -w
```

---

### Comprobar Usuarios con UID 0 Y GID 0 (SUPERUSUARIOS) creados:

---

```
SUPERUSUARIOS_UID=$(awk -F: '{if ($3==0) print $1}' /etc/passwd) ; echo $SUPERUSUARIOS_UID
SUPERUSUARIOS_GID=$(awk -F: '{if ($3==0) print $1}' /etc/passwd) ; echo $SUPERUSUARIOS_GID
```

---

### Comprobar estado de la Memoria RAM y Swap:

---

```
MEM_TOTAL=$(free -h | sed '1 d' | grep Mem: | awk '{print $2}') ; echo $MEM_TOTAL
MEM_USADA=$(free -h | sed '1 d' | grep Mem: | awk '{print $3}') ; echo $MEM_USADA
MEM_LIBRE=$(free -h | sed '1 d' | grep Mem: | awk '{print $4}') ; echo $MEM_LIBRE
MEM_COMPARTIDA=$(free -h | sed '1 d' | grep Mem: | awk '{print $5}') ; echo $MEM_COMPARTIDA
MEM_ALMACENADA=$(free -h | sed '1 d' | grep Mem: | awk '{print $6}') ; echo $MEM_ALMACENADA
MEM_CACHEADA=$(free -h | sed '1 d' | grep Mem: | awk '{print $7}') ; echo $MEM_CACHEADA
SWAP_TOTAL=$(free -h | sed '1 d' | grep Swap: | awk '{print $2}') ; echo $SWAP_TOTAL
SWAP_USADA=$(free -h | sed '1 d' | grep Swap: | awk '{print $3}') ; echo $SWAP_USADA
SWAP_LIBRE=$(free -h | sed '1 d' | grep Swap: | awk '{print $4}') ; echo $SWAP_LIBRE
```

---

### Comprobar Estado de las particiones / punto de montaje de un Disco SATA:

---

```
PART1_TOTAL=$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print $2}') ; echo $PART1_TOTAL
PART1_USADO=$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print $3}') ; echo $PART1_USADO
PART1_DISPONIBLE=$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print $4}') ; echo $PART1_DISPONIBLE
PART1_PORCENTAJE=$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print $5}') ; echo $PART1_PORCENTAJE
PART1_PUNTOMONTAJE=$(df -h | sed '1 d' | grep /dev/sda5 | awk '{print $6}') ; echo $PART1_PUNTOMONTAJE
```

---

### Comprobar la Carga promedio del Sistema (Procesos encolados):

---

```
CARGA_1MIN=$(top -n 1 -b | grep "load average:" | awk '{print $10}' | sed 's/,//2') ; echo $CARGA_1MIN
CARGA_5MIN=$(top -n 1 -b | grep "load average:" | awk '{print $11}' | sed 's/,//2') ; echo $CARGA_5MIN
CARGA_15MIN=$(top -n 1 -b | grep "load average:" | awk '{print $12}' | sed 's/,//2') ; echo $CARGA_15MIN
CARGA_1MIN=$(uptime | awk '{print $8}' | sed 's/,//2') ; echo $CARGA_1MIN
CARGA_5MIN=$(uptime | awk '{print $9}' | sed 's/,//2') ; echo $CARGA_5MIN
CARGA_15MIN=$(uptime | awk '{print $10}' | sed 's/,//2') ; echo $CARGA_15MIN
```

---

### Comprobar procesos Zombies en el Sistema Operativo:

---

```
PROC_ZOMBIE=$(top -n 1 -b | grep "zombie" | awk '{print $10}') ; echo $PROC_ZOMBIE
```

---

### Comprobar el Tiempo de Total de Trabajo (Arranque/Encendido):

---

```
-----  
TIEMPO_ENCENDIDO=$(uptime | awk '{print $3,$4}' | cut -f1 -d,) ; echo $TIEMPO_ENCENDIDO
```

### **Comprobar parámetros de la tarjeta de Vídeo:**

```
=====
```

### **Fabricante:**

```
FAB_TVIDEO=$(lspci -v | grep "VGA" | cut -d " " -f05) ; echo $FAB_TVIDEO
```

```
=====
```

### **Memoria RAM:**

```
MEM_TVIDEO=$(lspci -v -s `lspci | awk '/VGA/{print $1}'` | sed -n '/Memory.*, prefetchable/s/.*\[size=\([^\]]+\)\]M\]/\1/p') ; echo $MEM_TVIDEO
```

```
=====
```

### **Modulo (Driver):**

```
DRV_TVIDEO=$(lspci -nnk | grep -i vga -A3 | grep 'in use' | cut -d " " -f05) ; echo $DRV_TVIDEO
```

```
=====
```

### **Aceleración 3D:**

```
A3D_TVIDEO=$(glxinfo | grep "direct rendering: Yes" | awk '{print $3}') ; echo $A3D_TVIDEO
```

### **Comprobar parámetros del Procesador (CPU):**

```
=====
```

### **Fabricante:**

```
FABRICANTE_CPU=$(grep "vendor_id" /proc/cpuinfo | sed q | awk '{print $3}') ; echo $FABRICANTE_CPU
```

```
=====
```

### **Modelo:**

```
MODELO_CPU=$(grep "model name" /proc/cpuinfo | sed q | cut -d ":" -f 2 | awk '{print $0}') ; echo $MODELO_CPU
```

```
=====
```

### **Cantidad:**

```
NUM_CPU=$(grep "processor" /proc/cpuinfo | sort -r | sed q | awk '{print $3}') ; TOTAL_CPU=$((`expr $NUM_CPU + 1`)) ; echo $TOTAL_CPU
```

```
=====
```

### **Núcleos por CPU:**

```
NUCLEO_CPU=$(grep "cpu cores" /proc/cpuinfo | sed q | awk '{print $4}') ; echo $NUCLEO_CPU
```

```
=====
```

## Total de Nucleos por CPUs:

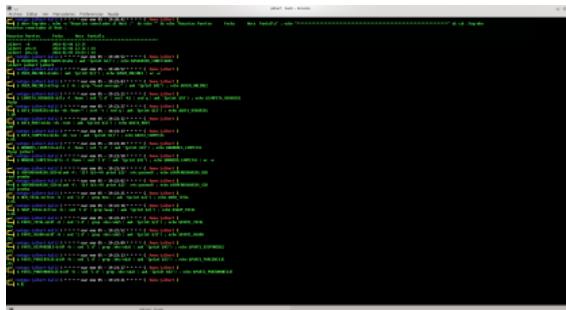
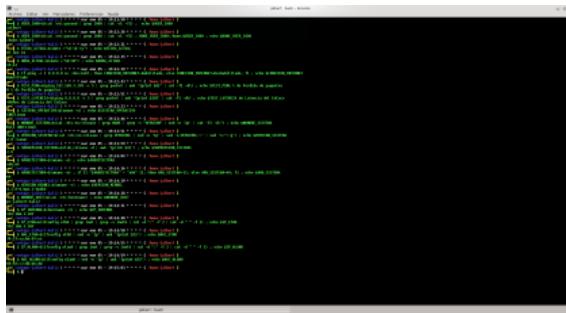
```
NUM_CPU=$(grep "processor" /proc/cpuinfo | sort -r | sed q | awk '{print $3}') ; TOTAL_CPU=$((`expr $NUM_CPU + 1`)) ; NUCLEO_CPU=$(grep "cpu cores" /proc/cpuinfo | sed q | awk '{print $4}') ; TOTAL_NUCLEO_CPU=$((`expr $TOTAL_CPU \* $NUCLEO_CPU`)) ; echo $TOTAL_NUCLEO_CPU
=====
=====
```

## Memoria caché del CPU:

```
CACHE_CPU=$(grep "cache size" /proc/cpuinfo | sed q | cut -d ":" -f 2 | awk '{print $0}') ; echo $CACHE_CPU
```

Espero que estos pequeños **“Tips”** les faciliten realizar optimizaciones básicas pero útiles, que por lo general solo están reservadas para personas expertas en **Tecnología, Informática, Software Libre y GNU/Linux**.

### PANTALLAZOS DE EJEMPLOS



**RECORDATORIO:** Si alguna línea de comando no se le ejecuta o muestra el valor correcto, asegúrese de probar manualmente cada sección de la orden de comando para intentar ajustar los valores o sintaxis de algunas variables.

**Te puede gustar**

Enlaces Patrocinados por Taboola

**¿Está tu hijo seguro online?**

Qustodio

---

**People from Venezuela cannot believe these flight prices**

Trips Shop

---

**¿Crees que conoces los motores de los coches? ¡Ponte a prueba con el ...**

Zoo Quizzes

---

**¡Últimas ofertas! Vuelos a partir de 690 Bs.F desde Nueva Esparta**

jetcost.co.ve

---

## Choose a plane and play this Game for 1 Minute

Delta Wars

---

## These Twins Were Named "Most Beautiful In The World," Wait Till You Se...

Give It Love

---

## TV Was Changed Forever By This Final MASH Episode

BuzzNet

---

## Hotels In Venezuela At Ridiculously Low Prices

Trips Insider

---

### Noticias Relacionadas

por Taboola

[Impresionante lista de aplicaciones y herramientas para Ubuntu / Linux](#)

[Guía de instalación de Voyager Linux 18.04 GS LTS](#)

[Convierte tu GNU/Linux en una Distro apta para el Desarrollo de Software](#)

[5 distribuciones de Linux para darle una nueva vida a tu equipo de bajos recursos](#)

## Categorías

[GNU/Linux, Tutoriales/Manuales/Tips](#)



## Linux Post Install

Ingeniero en Informática y Administrador de Servidores GNU/Linux con certificación internacional de Linux: 1.- Certified Linux Administrator (CLA) 2.- Certified Linux Operator (CLI)

[Se libera Bluefish 2.2.7 estable](#)  
[Extraer archivos tar a un directorio específico en Linux](#)

## 7 comentarios

---

1.  [Eduardo](#) dijo  
hace 3 años

Muchas gracias.

Muuy util.

[Responder a Eduardo](#)

2.  [HO2Gi](#) dijo  
hace 3 años

Muy bueno , gracias .

[Responder a HO2Gi](#)

3.  [Ing. Jose Albert](#) dijo  
hace 3 años

Un placer colaborar! Pronto montaré muchos más sobre el uso del Shell Scripting en casos muy prácticos.

[Responder a Ing. Jose Albert](#)

4.  [cr0t0](#) dijo  
hace 3 años

Muy bueno ingeniero! Ojala haya más articulos de Shell Scripting.

[Responder a cr0t0](#)

5.  [Francisco Tovar](#) dijo

hace 3 años

Excelentes los comandos allí mostrados y de gran utilidad.

[Responder a Francisco Tovar](#)



6.  [userarch](#) dijo

hace 3 años

Gracias por la valiosa información; pero en mi caso algunos comandos no dan el resultado esperado, por ejemplo el comando "who" y "w" no muestran nada; esto me ocurrió desde que hice alguna actualización del sistema(uso archlinux con administrador de pantalla "lxdm" y entorno gráfico "xfce 4.12"). Alguna idea de lo que ocurre(el resultado es el mismo aun si uso user root).

Gracias.

[Responder a userarch](#)



7.  [Ing. Jose Albert](#) dijo

hace 3 años

Les gustaría un Sistema experto hecho con Shell Scripting que a solicitud les generará la extracción de todos los parámetros del sistema en forma de un Informe.

Un ejemplo de lo que se puede hacer con Shell Scripting:

ScreenCast de Testeo del LPI-SB8 (LINUX POST INSTALL - SCRIPT BICENTENARIO 8.0.0)  
(lpi\_sb8\_adecuación-audiovisual\_2016.sh / 43Kb)

Ver Screencast: <https://www.youtube.com/watch?v=cWpVQcbgCyY>

[Responder a Ing. Jose Albert](#)

## Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con \*

Comentario

Nombre\*

Correo electrónico\*



Acepto los términos de privacidad\*

1. Responsable de los datos: Miguel Ángel Gatón
2. Finalidad de los datos: Controlar el SPAM, gestión de comentarios.
3. Legitimación: Tu consentimiento
4. Comunicación de los datos: No se comunicarán los datos a terceros salvo por obligación legal.
5. Almacenamiento de los datos: Base de datos alojada en Occentus Networks (UE)
6. Derechos: En cualquier momento puedes limitar, recuperar y borrar tu información.



Quiero recibir la newsletter

- [Facebook](#)
- [Twitter](#)
- [Google+](#)
- [Email](#)
- [RSS](#)

## Noticias en tu email

Recibe las últimas noticias sobre Linux en tu correo electrónico

Nombre

Email

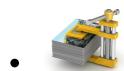
Newsletter diaria

Newsletter semanal

[Acepto las condiciones legales](#)

[Suscribir](#)

## TOP artículos



- [Cómo comprimir y descomprimir ficheros en Linux](#)

### Secciones

[Android](#)

[Apariencia/Personalización](#)

[Aplicaciones](#)

[ArchivoUsemosLinux](#)

[DesdeLinux](#)

[Diseño](#)

[Distribuciones](#)

[Featured](#)

[GNU/Linux](#)

[Juegos](#)

[Noticias](#)

[Otras](#)

[Programación](#)

[Recomendados](#)

[Redes/Servidores](#)

**Tutoriales/Manuales/Tips**

**Varios**

**Wordpress**

- [Plugins Wordpress](#)
- [Tutoriales Wordpress](#)

**Índice del artículo**

[1 Fabricante:](#)

[2 Memoria RAM:](#)

[3 Modulo \(Driver\):](#)

[4 Aceleración 3D:](#)

[5 Fabricante:](#)

[6 Modelo:](#)

[7 Cantidad:](#)

[8 Nucleos por CPU:](#)

[9 Total de Nucleos por CPUs:](#)

[10 Memoria caché del CPU:](#)

↑

- [Secciones](#)
- [Aplicaciones](#)
- [Distribuciones](#)
- [Comunidades](#)
- [Repositorio de tips](#)
- [Suscribir newsletter](#)
- [Aviso Legal](#)
- [Equipo editorial](#)
- [Contacto](#)