



EBook Gratis

APRENDIZAJE shell

Free unaffiliated eBook created from
Stack Overflow contributors.

#shell

Tabla de contenido

Acerca de	1
Capítulo 1: Empezando con shell	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Hola Mundo.....	3
Capítulo 2: Descargando el último Artífcat de Artifactory usando Shell Script	5
Introducción.....	5
Examples.....	5
PASOS PARA DESCARGAR EL ÚLTIMO ARTEFACTO.....	5
Capítulo 3: Diferente formato de fecha / hora en shell	7
Parámetros.....	7
Observaciones.....	8
Examples.....	8
Código de ejemplo y salida.....	8
Salida	8
Capítulo 4: usermod	10
Examples.....	10
agregar un usuario a un grupo.....	10
Creditos	11

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [shell](#)

It is an unofficial and free shell ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official shell.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con shell

Observaciones

Esta sección proporciona una descripción general de qué es el shell y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro del shell y vincular a los temas relacionados. Dado que la Documentación para shell es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Un shell de comandos es un programa de computadora de interfaz de línea de comandos para un sistema operativo.

Algunas variantes

1. **Bash** : viene como shell predeterminado en ubuntu

2. **KornShell (ksh)** :

Para instalar ksh en Ubuntu

```
$ sudo apt-get install ksh
```

Para empezar a trabajar con ksh

```
$ ksh
$ ps $$
  PID TTY          STAT       TIME COMMAND
  4187 pts/2    S           0:00   ksh
```

Ingrese los comandos en el indicador de ksh

3. **csH** :

Para instalar csh en Ubuntu

```
$ sudo apt-get install csh
```

Para trabajar con csh, vaya a la línea de comando e ingrese csh

```
$ csh
%
```

Hola Mundo

En el símbolo del sistema:

```
$ echo "Hello World"
```

Salida:

```
Hello World
```

Para crear un script, cree un documento de texto con el siguiente contenido:

```
#!/bin/sh  
echo "Hello World"
```

Guarde el script con el nombre `hello.sh` (o cualquier nombre de archivo) y haga que el script sea ejecutable otorgando el siguiente permiso:

```
$ chmod 755 hello.sh
```

O:

```
$ chmod +x hello.sh
```

Ejecutar el script:

```
$ ./hello.sh
```

Salida:

```
Hello World
```

Para ejecutar un script de shell local sin permiso ejecutable:

1.bash

```
$ bash hello.sh  
Hello World
```

2.ksh

```
$ ksh hello.sh  
Hello World
```

3.sh

```
$ sh hello.sh
```

Hello World

Lea Empezando con shell en línea: <https://riptutorial.com/es/shell/topic/1068/empezando-con-shell>

Capítulo 2: Descargando el último Artífcat de Artifactory usando Shell Script

Introducción

Descargue el último artefacto del repositorio de artefactos usando un script de shell.

Examples

PASOS PARA DESCARGAR EL ÚLTIMO ARTEFACTO

1. Obteniendo la respuesta JSON para el último artefacto modificado (el último)

```
latestArtifactUriResponse=curl -u username:password --silent
https://hostname.com/artifactory/api/storage/<repo_name>/<folder_name>/?lastModified | grep
uri | awk '{ print $3 }' | sed s/\\"//g | sed s/,//g
```

Se devolverá una respuesta en el siguiente formato:

```
{
  "uri" :
  "https://hostname.com/artifactory/api/storage/<repo_name>/<folder_name>/latest_artifact.tar.gz",
  "lastModified" : "2016-12-22T04:26:25.534-0500"
}
```

2. Recuperando la URL directa al último artefacto desde "latestArtifactUriResponse"

El `latestArtifactUriReponse` devolverá la respuesta en el siguiente formato:

```
{
  "repo" : "repo_name",
  "path" : "/folder_name/latest_artifact.tar.gz",
  "created" : "2016-12-22T04:26:29.482-05:00",
  "createdBy" : "username",
  "lastModified" : "2016-12-22T04:26:25.534-05:00",
  "modifiedBy" : "username",
  "lastUpdated" : "2016-12-22T04:26:25.534-05:00",
  "downloadUri" :
  "https://hostname.com/artifactory/repo_name/folder_name/latest_artifact.tar.gz",
  "mimeType" : "application/octet-stream",
  "size" : "94310686",
  "checksums" : {
    "sha1" : "ocb778e566890b0f3d115b828ce8dd4e840",
    "md5" : "d050fb8108745973cf0d64e15667b340"
  },
  "originalChecksums" : {
  },
  "uri" :
  "https://hostanme.com/artifactory/api/storage/repo_name/folder_name/latest_artifact.tar.gz"
```

```
}
```

Aquí tenemos que obtener el valor de `downloadUri` que es la URL directa del último artefacto y almacenarlo en la variable `downloadUrl` :

```
downloadUrl=`curl -u username:password --silent $latestArtifactUrl | grep downloadUri | awk '{
print $3 }' | sed s/\\"//g | sed s/,//g`
```

3. Descargando el último artefacto

```
curl -u username:password -O $downloadUrl
```

Y ya está hecho.

Lea [Descargando el último Artefacto de Artifactory usando Shell Script en línea](https://riptutorial.com/es/shell/topic/8884/descargando-el-ultimo-artifact-de-artifactory-usando-shell-script):

<https://riptutorial.com/es/shell/topic/8884/descargando-el-ultimo-artifact-de-artifactory-usando-shell-script>

Capítulo 3: Diferente formato de fecha / hora en shell

Parámetros

Formato	Interpretado como
%%	signo de porcentaje literal (%)
%A	Nombre del día de la semana (por ejemplo, domingo)
%a	Nombre del día de la semana en formato corto (ej. Sol)
%B	nombre del mes completo (por ejemplo, enero)
%b	nombre del mes (por ejemplo, enero)
%H	hora (00..23)
%I	hora (01..12)
%j	día del año (001..366)
%k	hora (0..23)
%l	hora (1..12)
%M	minuto (00..59)
%m	mes (01..12)
%p	Definir AM o PM; en blanco si no se conoce
%R	Hora y minuto de 24 horas; igual que% H:% M
%r	Reloj de 12 horas (por ejemplo, 11:11:04 PM)
%S	segundo (00..60)
%s	Época de Unix: segundos desde 1970-01-01 00:00:00 UTC (<i>no disponible en UNIXes más antiguos</i>)
%T	tiempo, equivalente a %H:%M:%S
%Z	nombre de la zona horaria (por ejemplo, PDT)
%z	desplazamiento de zona horaria (dirección, horas, minutos, por ejemplo, -0700)

Observaciones

A continuación se muestran algunos enlaces útiles para el comando `date` en shells de Unix:

- Linux: la [página de manual de GNU para la fecha](#) , incluye códigos de formato, vea también la [página de manual de GNU para strftime](#)
- FreeBSD: [página de manual de BSD para la fecha](#) , códigos de formato en vivo en la [página de manual de BSD para strftime](#)
- Apple: [página de manual de OS X para la fecha](#) , códigos de formato en vivo en la [página de manual de OS X para strftime](#)
- Época: [tiempo de Unix](#) , también conocido como tiempo POSIX (`%s` , segundos desde 1970)

Examples

Código de ejemplo y salida

```
#!/bin/bash

#Print Date / Time in different Formats
date1=$(date +%d-%m-%y)
date2=$(date +%d-%m-%Y)
date3=$(date +%d-%b-%Y)
date4=$(date +%d-%B-%Y)
date5=$(date +%a %d-%b-%Y)
date6=$(date +%a %d-%b-%Y %Z)
date7=$(date +%A %d-%b-%Y)

echo "Print Date in different format"
echo $date1
echo $date2
echo $date3
echo $date4
echo $date5
echo $date6
echo $date7
echo

#print Timestamp
time1=$(date +%H:%M:%S)
time2=$(date +%I:%M:%S)
time3=$(date +%r)
time4=$(date +%R)

echo "Print Time in different format"
echo "Time in 24h clock: $time1"
echo "Time in 12h clock: $time2"
echo "Time with AM/PM: $time3"
echo "Time in hour&minute: $time4"

exit
```

Salida

```
Print Date in different format
```

```
01-08-16
```

```
01-08-2016
```

```
01-Aug-2016
```

```
01-August-2016
```

```
Mon 01-Aug-2016
```

```
Mon 01-Aug-2016 IST
```

```
Monday 01-Aug-2016
```

```
Print Time in different format
```

```
Time in 24h clock: 15:16:06
```

```
Time in 12h clock: 03:16:06
```

```
Time with AM/PM: 03:16:06 PM
```

```
Time in hour&minute: 15:16
```

Lea Diferente formato de fecha / hora en shell en línea:

<https://riptutorial.com/es/shell/topic/4850/diferente-formato-de-fecha---hora-en-shell>

Capítulo 4: usermod

Examples

agregar un usuario a un grupo

digamos que tiene dos usuarios con el nombre de usuario `chyingp` y `casper`, y desea agregarlos a ambos para agrupar la `root`.

```
usermod chyingp -a -G root
usermod casper -a -G root
```

Aquí viene la explicación de `-a`, `-G`

`-a, --append`

Add the user to the supplementary group(s). Use only with the `-G` option.

`-G, --groups GROUP1[,GROUP2,...[,GROUPN]]`

A list of supplementary groups which the user is also a member of. Each group is separated from the next by

a comma, with no intervening whitespace. The groups are subject to the same restrictions as the group given

with the `-g` option.

If the user is currently a member of a group which is not listed, the user will be removed from the group.

This behaviour can be changed via the `-a` option, which appends the user to the current supplementary group

list.

Lea usermod en línea: <https://riptutorial.com/es/shell/topic/6141/usermod>

Creditos

S. No	Capítulos	Contributors
1	Empezando con shell	agc , Ankur Anand , Community , juleslasne , Kusalananda , Lee HoYo , mklement0
2	Descargando el último Artificat de Artifactory usando Shell Script	ANIL
3	Diferente formato de fecha / hora en shell	Adam Katz , UUU
4	usermod	casperchen