FREE eBook

LEARNING shell

Free unaffiliated eBook created from
Stack Overflow contributors.

#shell

Table of Contents

About

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.

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

Chapter 1: Getting started with shell

Remarks

This section provides an overview of what shell is, and why a developer might want to use it.

It should also mention any large subjects within shell, and link out to the related topics. Since the Documentation for shell is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

A command shell is a command line interface computer program to an operating system.

Some Variants

1. *Bash* : Comes as default shell on ubuntu

2. KornShell(ksh) :

To install ksh in Ubuntu

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

To start working with ksh

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

Enter the commands at the ksh prompt

3. *csh* :

To install csh in Ubuntu

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

To work with csh, go to the command line and enter csh

```
$ csh
%
```

Hello World

At the command prompt:

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

Output:

```
Hello World
```

To create a script, create a text document with the following content:

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

Save the script with the name `hello.sh` (or any filename) and make the script executable by giving the following permission:

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

Or:

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

Run the script:

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

Output:

```
Hello World
```

To run a local shell script without executable permission:

1.bash

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

2.ksh

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

3.sh

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

Read Getting started with shell online: https://riptutorial.com/shell/topic/1068/getting-started-with-shell

Chapter 2: Different Date/Time format in shell

Parameters

Format	Interpreted as
`%%`	literal percent sign (%)
`%A`	weekday name (e.g. Sunday)
`%a`	weekday name in short fomat (e.g. Sun)
`%B`	full month name (e.g. January)
`%b`	month name (e.g. Jan)
`%H`	hour (00..23)
`%I`	hour (01..12)
`%j`	day of year (001..366)
`%k`	hour (0..23)
`%l`	hour (1..12)
`%M`	minute (00..59)
`%m`	month (01..12)
`%p`	Define AM or PM; blank if not known
`%R`	24-hour hour and minute; same as %H:%M
`%r`	12-hour clock time (e.g. 11:11:04 PM)
`%S`	second (00..60)
`%s`	Unix epoch: seconds since 1970-01-01 00:00:00 UTC *(not available in older UNIXes)*
`%T`	time, equivalent to `%H:%M:%S`
`%Z`	time zone name (e.g. PDT)
`%z`	time zone offset (direction, hours, minutes, e.g. -0700)

Remarks

Below are a few useful links for the `date` command in Unix shells:

- Linux: GNU man page for date, includes format codes, see also GNU man page for strftime
- FreeBSD: BSD man page for date, format codes live in BSD man page for strftime
- Apple: OS X man page for date, format codes live in OS X man page for strftime
- Epoch: Unix time, also known as POSIX time (`%s`, seconds since 1970)

Examples

Sample Code & output

```
#!/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
```

Output

```
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
```

Read Different Date/Time format in shell online: https://riptutorial.com/shell/topic/4850/different-date-time-format-in-shell

Chapter 3: Downloading Latest Artifcat from Artifactory using Shell Script

Introduction

Download latest Artifact from Artifactory repository using shell script.

Examples

STEPS TO DOWNLOAD THE LATEST ARTIFACT

1. Fetching the JSON response for last modified (latest) artifact

```
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
```

It will return a response in the following format:

```
{
  "uri" :
"https://hostname.com/artifactory/api/storage/<repo_name>/<folder_name>/latest_artifact.tar.gz",

  "lastModified" : "2016-12-22T04:26:25.534-0500"
}
```

2. Fetching the direct URL to the latest artifact from the "latestArtifactUriResponse"

The `latestArtifactUriReponse` will return the response in the following format:

```
{
  "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"
```

```
}
```

Here we have to fetch the value of `downloadUri` which is the direct URL to the latest artifcat and store it in `downloadUrl` variable:

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

3. Downloading the latest artifact

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

And you are done.

Read Downloading Latest Artifcat from Artifactory using Shell Script online:
https://riptutorial.com/shell/topic/8884/downloading-latest-artifcat-from-artifactory-using-shell-script

Chapter 4: usermod

Examples

add a user to a group

let's say you have two users with username `chyingp` and `casper`, and want to add them both to group `root`.

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

here comes the explanation of `-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.
```

Read usermod online: https://riptutorial.com/shell/topic/6141/usermod

Credits

S. No	Chapters	Contributors
1	Getting started with shell	agc, Ankur Anand, Community, juleslasne, Kusalananda, Lee HoYo, mklement0
2	Different Date/Time format in shell	Adam Katz, UUU
3	Downloading Latest Artifcat from Artifactory using Shell Script	ANIL
4	usermod	casperchen