

僕的備忘録 ~なんかいろいろ~

Linux・Windows・Mac・ネットワーク・プログラム等をメインに書いています



Home Linux Mac Windows コマンド Tool まとめ 質問



覚えてると案外便利なBashのリダイレクト・パイプの使い方9個

© 2016年2月29日 □ Apple ☁ 1 Comment

3 Like 0

ツイート

G+ Pocket

なんなしにBashのリダイレクトとパイプについて調べてたので、後から確認しやすいようにまとめてみることにする。

目次 [hide]

- 1.ファイルへのリダイレクト
- 2.ファイルの読み込み
- 3.複数行の文字列をコマンドに渡す
- 4.コマンドに文字列をリダイレクトする
- 5.execコマンドと組み合わせる
- 6.グループ化してまとめてリダイレクトする
- 7.コマンドに複数コマンドの実行結果を渡す
- 8.コマンドの実行結果を他のコマンドに連携する(パイプ)
- 9.パイプでつなげた各コマンドの実行結果(exitcode)を確認する

1.ファイルへのリダイレクト

基本的な使い方。コマンドの実行結果をファイルにリダイレクトする事で、上書きや追記をさせることが出来る。
これについては、UNIX系OSに慣れていない人でもよく知っている事だろう。
「>」で上書き(ファイルの既存内容は削除される)、「>>」で追記書き込みが行われる。

```
1 | コマンド > ファイル # 上書き
2 | コマンド >> ファイル # 追記
```

Sponsored Links

① ×

楽天証券で投資デビュー

楽天証券で毎月100円
 投資デビュー！手軽に
 手間なく申込完了。お
 手軽な投資信託も充実

To search type and hit enter

最近の投稿

- バックスペースが含まれるテキストを、ターミナル以外でもちゃんと表示できるよう加工する 2018年4月17日
- コンソール上でアルファベット・数字の半角⇒全角変換をする 2018年4月16日
- シェル芸で数字を単位付き(万億兆垓)にして読みやすくする 2018年4月12日
- Linuxでプロセスが開こうとしているファイルを調べる 2018年4月11日
- 第35回シェル芸勉強会に参加してきました(復習) 2018年4月10日

カテゴリ

Apple awk CentOS CUI
 Debian GaleraCluster Gentoo
 Graylog grep GUI HP-UX Linux
 Mac MariaDB Microsoft
 NixOS PowerShell Proxmox Python
 sed ShellScript Solaris ssh Teraterm
 Tool Ubuntu UNIX VMware
 ESXi Vyatta Vyos Webサービス Webベース
 Windows その他 まとめ クラウド □
 マンド セキュリティ データベース ネットワーク プログラミング 言語 ログ管理 仮想 監視 豆知識

よく見られているページ

- LAN内で使われているIPアドレスを調べる6つの方法 (Linux/Windows)
- これだけは最低限おさえておきたいTeratermの便利な機能8個
- Linux/UNIXで文字列から特定部分(右から・左から何個、○○～○○まで)を抜き出すコマンド
- findコマンドで覚えておきたい使い方12個(+1個)

で、このリダイレクトだが通常は標準出力しかリダイレクトされない。
この辺りを使い分ける場合は、以下のようにコマンドを実行する。
※追記書き込み時も同様のため割愛。

```
1 | コマンド 1> ファイル # 標準出力のみをリダイレクト(デフォルト)
2 | コマンド 2> ファイル # 標準エラー出力のみをリダイレクト
3 | コマンド &> ファイル # 標準出力・標準エラー出力を共にリダイレクト
```

これを利用することで、以下のように標準エラー出力だけを/dev/nullにリダイレクトしてコンソールにもファイルにも出力させないようにしたり、それぞれを別のファイルに出力させたりすることができる。
ちょっとしたお遊びとしては、他のコンソールに任意の文字列を送りつけるといった事も可能だ。たとえば、以下のようにコマンドを実行することで「/dev/pts/1」を使っているコンソールに文字列を送ることができる。

```
1 | echo "文字列" > /dev/pts/1
```

ちなみに、標準エラー出力と標準出力を入れ替える場合は、以下のようにコマンドを実行する事で行える。

```
1 | コマンド 3>&1 1>&2 2>&3
```

2.ファイルの読み込み

リダイレクトを使うことで、ファイルを一部のコマンドに読み込ませる事も出来る。

```
1 | コマンド < ファイル # ファイルの内容を一部のコマンドに読み込ませる
```

以下、readコマンドを用いた例

```
[root@test-node ~]# cat /work/command
/etc
[root@test-node ~]# read -r line < /work/command ;echo $line
/etc
```

3.複数行の文字列をコマンドに渡す

これもいろんな所で見る使い方だ。
以下のようにコマンドを実行することで、複数行の文字列をコマンドにリダイレクトすることが出来る。

```
1 | コマンド << EOF
2 | .....
3 | EOF
```

よく使われるには、catと組み合わせてファイルに書き込みをする方法だろう。
以下、例。

```
[root@test-node ~]# cat << EOF > /work/test_red.txt
> 1234
> 5678
> 90
> EOF
[root@test-node ~]# cat /work/test_red.txt
1234
5678
90
```

ここで記述されているEOFというのは処理終了の判定文字列となる。
ここではEOFと記述しているが、他の文字列でも構わない。

なお、Tabなどを入力したい場合は、以下のようにコマンドを実行する。

```
1 | コマンド <<- EOF
2 | .....
3 | EOF
```



Linux/UNIXでファイルの文字コード(UTF-8 or Shift JIS or EUC-JP...)を確認する



grepコマンドで覚えておきたい使い方14個(+3個)



Windowsで容量の大きいファイル&ディレクトリを調べる



CentOS7にSambaをインストール



sedコマンドで覚えておきたい使い方12個(+3個)



topコマンドで覚えておきたい使い方14個

Twitterタイムライン

@blacknon_さんのツイート



Blacknon@通勤電車リハビリ
@blacknon_

電車随分遅れたな...
想定より20分くらい駅の到着時間が遅れてしまった

8時間



Blacknon@通勤電車リハビリ
@blacknon_

シェル芸bot、Golangで書かれてたんだ

8時間



Blacknon@通勤電車リハビリ
@blacknon_

これがgithub.com/theoldmoon0602...

theoldmoon0602/...
シェル芸を実行しちゃう
github.com

8時間



Blacknon@通勤電車リハビリ
@blacknon_

埋め込む Twitterで表示

4.コマンドに文字列をリダイレクトする

以下のように指定することで、コマンドに文字列をリダイレクトする事も出来る。

```
1 | コマンド <<< "文字列"
```

例えば、catコマンドに文字列を送ることでechoと同じような処理を行わせる事ができる。
(日常で使うかわからないけど)

```
[root@test-node ~]# cat <<< "This Server is $(hostname)."
This Server is test-node.
```

5.execコマンドと組み合わせる

execコマンド(forkせずにコマンドを実行する)と組み合わせる事で、エラーのリダイレクト設定をそのセッション内では継続させるような事もできる。
(ただし、PS1などがコンソールに出力されない状態になる場合もあるので、シェルスクリプト内でしか使えないと思われる)

以下、簡単にまとめ。

- セッション中は永続的に標準エラー出力を指定したファイルにリダイレクトする

```
1 | exec 2> ファイル # セッション中は永続的に標準エラー出力を指定したファイルにリダイレクトする
```

```
[root@test-node ~]# cat /work/test_sh.sh
exec 2>/tmp/error_test
grep
grep 'aaa' /etc/hogehogege
[root@test-node ~]# sh /work/test_sh.sh
[root@test-node ~]# cat /tmp/error_test
/work/test_sh.sh: 行 2: grep: コマンドが見つかりません
grep: /etc/hogehogege: そのようなファイルやディレクトリはありません
```

- ファイルを読み込む

```
1 | exec 3< ファイル
```

```
[root@test-node ~]# exec 3</etc/fstab
[root@test-node ~]# grep "" <&3
#
# /etc/fstab
# Created by anaconda on Sun Jan 3 08:46:04 2016
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root / xfs defaults 0 0
UUID=461b7464-714d-4a8b-9f63-19fe0482887f /boot xfs defaults 0 0
/dev/mapper/centos-home /home xfs defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
[root@test-node ~]# grep "" <&3
[root@test-node ~]# </pre>
```

- ファイルへ書き込む

```
1 | exec 3> ファイル
```

```
[root@test-node ~]# exec 3>/tmp/3teest
[root@test-node ~]# echo "aaaaaaaa" >&3
```



```
[root@test-node ~]# cat /tmp/3teest
aaaaaaaaa
```

●ファイルへ書き込み・読み込みを行う

```
1 | exec 3<> ファイル
```

```
[root@test-node ~]# exec 3<>/tmp/3test
[root@test-node ~]# echo "aaaaaaaaa" >&3
[root@test-node ~]#
[root@test-node ~]# exec 3<>/tmp/3test
[root@test-node ~]# grep "" <&3
aaaaaaaaa
```

●指定ポートに対し通信する

```
1 | exec 3<> /dev/tcp/アドレス/ポート番号 # tcpの場合
2 | exec 3<> /dev/udp/アドレス/ポート番号 # udpの場合
```

```
[root@test-node ~]# exec 3<> /dev/tcp/127.0.0.1/22
[root@test-node ~]# echo $?
0
[root@test-node ~]# exec 3<> /dev/tcp/127.0.0.1/23
-bash: connect: 接続を拒否されました
-bash: /dev/tcp/127.0.0.1/23: 接続を拒否されました
[root@test-node ~]# echo $?
1
```



6.グループ化してまとめてリダイレクトする

前にも[ここ](#)で書いた事があるが、「()」もしくは「{}」でコマンドを囲むことでまとめてリダイレクトさせることが出来る。

```
1 | (コマンド1;コマンド2;...) > ファイル # サブシェルあり
2 | { コマンド1;コマンド2;...; } > ファイル # サブシェルなし
```

```
[root@test-node ~]# (echo aaa;echo bbb) > /tmp/aaa123
[root@test-node ~]# cat /tmp/aaa123
aaa
bbb
[root@test-node ~]#
[root@test-node ~]# { echo aaa;echo bbb; } > /tmp/aaa123_2
[root@test-node ~]# cat /tmp/aaa123_2
aaa
bbb
```

7.コマンドに複数コマンドの実行結果を渡す

一つのコマンドの実行結果を渡すだけなら「|(パイプ)」でいいのだが、時には複数コマンドの実行結果を渡したい時だつてある。
 その場合は、シェル芸勉強会などではよく見かけるのだが、以下のようにリダイレクトさせればよい。
 (一応、この機能には「Process Substitution」という名前が付いているらしい)

1 | コマンド <(コマンド)

複数ある場合は、そのままスペース区切りで続けて記述すれば良いだけだ。

1 | コマンド <(コマンド) <(コマンド)

```
[root@test-node ~]# cat /work/test_sort1
1 dddddd
1 aaaaaa
5 bbbbbbb
6 gfgfgr
[root@test-node ~]# cat /work/test_sort2
6 gfgfgr
4 bbbbbbb
1 aaaaaa
3 cfffff
[root@test-node ~]# sort /work/test_sort1
1 aaaaaa
2 dddddd
5 bbbbbbb
6 gfgfgr
[root@test-node ~]# sort /work/test_sort2
1 aaaaaa
3 cfffff
4 bbbbbbb
6 gfgfgr
[root@test-node ~]# diff <(sort /work/test_sort1) <(sort /work/test_sort2)
2,3c2,3
< 2 dddddd
< 5 bbbbbbb --- > 3 cfffff
> 4 bbbbbbb
```

8.コマンドの実行結果を他のコマンドに連携する(パイプ)

流石にこれは説明しなくてもいいだろう。
 コマンドの実行結果をパイプで繋げる事で連携していく、UNIX系OSのシェルなら大体持っている機能だ。

1 | コマンド | コマンド

```
[root@test-node ~]# grep "" /work/hostname_*
/work/hostname_aaa_000.txt:aaa_000
/work/hostname_aaa_001.txt:aaa_001
/work/hostname_aaa_002.txt:aaa_002
/work/hostname_aaa_003.txt:aaa_003
/work/hostname_aaa_004.txt:aaa_004
/work/hostname_aaa_005.txt:aaa_005
/work/hostname_bbb_000.txt:aaa_000
/work/hostname_bbb_001.txt:aaa_001
/work/hostname_bbb_002.txt:aaa_002
/work/hostname_bbb_003.txt:aaa_003
/work/hostname_bbb_004.txt:aaa_004
/work/hostname_bbb_005.txt:aaa_005
[root@test-node ~]# grep "" /work/hostname_* | grep aaa_000
/work/hostname_aaa_000.txt:aaa_000
/work/hostname_bbb_000.txt:aaa_000
```

なお、bash4.1以降の場合だと、以下のようにコマンドを実行することで標準エラー出力を渡す事も出来るようになつたようだ。

1 | コマンド |& コマンド

```
[root@test-node ~]# grep "" /work/hostneme_*
grep: /work/hostneme_*: そのようなファイルやディレクトリはありません
[root@test-node ~]#
[root@test-node ~]# grep "" /work/hostneme_* | sed 's/work/aaaa/g'
grep: /work/hostneme_*: そのようなファイルやディレクトリはありません
[root@test-node ~]#
```

```
[root@test-node ~]# grep "" /work/hostneme_* |& sed 's/work/aaaa/g'
grep: /aaaa/hostneme_*: そのようなファイルやディレクトリはありません
```

9.パイプでつなげた各コマンドの実行結果(exitcode)を確認する

パイプでつなげた場合、途中のコマンドでエラーが出てもecho \$?ではパイプの最後にあるコマンドの結果しか表示されない。
各コマンドのexitcodeを取得する場合は、bash変数\$PIPESTATUSから取得すると良いだろう。

```
1 | echo ${PIPESTATUS[@]}
```

```
[root@test-node ~]# grep "" /euc |& sed 's/euc/aaa/g'
grep: /aaa: そのようなファイルやディレクトリはありません
[root@test-node ~]# echo ${PIPESTATUS[@]}
2 0
```

覚えて便利 いますぐ使える! シェルスクリプトシンプルレシピ54

著者 松浦智之
 価格 ￥1,512 (2018/04/25 22:21時点)
 出版日 2011/08/10
 商品ランキング 175,917位
 単行本 122ページ
 ISBN-10 4904807049
 ISBN-13 9784904807040
 出版社 USP研究所

Like 0 ツイート G+ Pocket



か、会社のソフトが
不正コピーだった
にやんて・・・

The Software Alliance

不正コピーソフトを使用中の会社をご存じなら、
BSAにお知らせ下さい。有力情報には**最高で100万円**の
報奨金を差し上げます（2013年 実績は平均約25万円）。

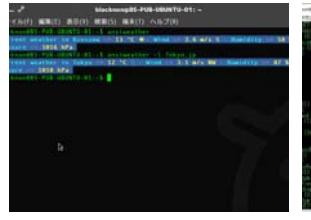
*報奨金の提供には一定の条件があります。[詳しく見る>>](#)

Written by blacknon



インフラエンジニア(...のつもり)。仕事で使うならクライアントはWindowsよりはUNIXの方が好き。大体いつも眠い。

Related Posts



コンソール上に一行のお天気情報を出力する
『AnsiWeather』
調べものしてたところ、コンソ...
ー



aproposコマンドで指定した
キーワードを含むmanを検索
する
今まで知らなかったのだが、『a...
ー



CentOS 7とSamba4で自宅用
Active Directory Domain
Controller (AD DC)を構築す
る①インストール編
一定の規模を持つWindows...
ー

This article has 1 comments

Pingback: [シェルスクリプト入門 中級編 | Full Stack Engineer](#)

[Leave a Comment](#)

メールアドレスが公開されることはありません。

(comment form area)

名前

(input field for name)

メールアドレス

(input field for email address)

ウェブサイト

(input field for website)



(input field for CAPTCHA code)

キャプチャコード *

コメントを送信

新しいコメントをメールで通知

新しい投稿をメールで受け取る

◎ CentOS 7にZabbix 3.0をインストールしてみる

Linux/UNIXで文字列から特定部分(右から・左から何個、○○~○○まで)を
抜き出すコマンド ◎

タグ

AIX Apple awk CentOS CUI Debian
GaleraCluster Gentoo Graylog grep HP-UX KVM
Linux Mac MariaDB Microsoft NixOS
PowerShell Proxmox Python sed ShellScript Solaris
SSH Teraterm Tool Ubuntu UNIX VMwareESXi
Vyatta VyOS Webベース Windows クラウド コマンド
セキュリティ データベース ネットワーク プログラミング言語 ログ管理 仮想
データベース ネットワーク プログラミング言語 ログ管理 仮想
豆知識

カテゴリー

Apple awk CentOS CUI Debian
GaleraCluster Gentoo Graylog grep GUI HP-UX
Linux Mac MariaDB Microsoft NixOS
PowerShell Proxmox Python sed ShellScript Solaris ssh
Teraterm Tool Ubuntu UNIX VMware ESXi
Vyatta VyOS Webサービス Webベース Windows その他
まとめ クラウド コマンド セキュリティ データ
ベース ネットワーク プログラミング言語 ログ管理 仮想
監視 豆知識

Powered by 僕的備忘録 ~なんかいろいろ~