

PDF 閲覧ソフトで「しおり」をクリックすると目次が表示されます。

あれば便利なシェルとエイリアス

(CentOS 版)

マニュアル

1.0 版

2012 年 5 月 24 日

ec4oss

<http://sourceforge.jp/projects/tools4centos/>

履歴

版数	日付	内容
1.0	2012/05/24	1.0 版として一式リリース

目 次

1	はじめに.....	4
2	導入方法.....	5
2.1	導入前の確認事項.....	5
2.2	導入方法	5
3	使用方法.....	8
3.1	dg コマンド	8
3.2	dgi コマンド.....	9
3.3	lu コマンド.....	10
3.4	lus コマンド.....	11
3.5	コマンドエイリアス	12
4	最後に.....	13
4.1	参考 Web サイト.....	13
4.2	バージョンについて	14
4.3	その他	14

1 はじめに

当資料は CentOS 用に開発されたオープンソースのシェル「あれば便利なシェル (CentOS 版)」(以降「当シェル群」と記します) の導入方法や使用方法等を記したものです。対象読者として「CentOS の基礎的な知識 (基本的なコマンド操作や環境変数 PATH の意味等) を持っておられる方」「CentOS で bash を使用される方 (csh は対象外)」を想定にしています。

例えば CentOS の任意のディレクトリ配下で「5 月 10 日以降に更新された全てのファイルを再帰検索し、更新日時を秒単位で表示する」という操作は、ものすごく複雑なコマンド入力が必要となります。当シェル群は、このような複雑な操作が簡単に行えるシェルやエイリアスです。4 個のシェルと 8 個のエイリアスが CentOS での構築・管理・運用を強力に支援します。(ブログ等で公開されている tips 等を参考に「使いやすさ重視」でまとめました。CentS5.5 および CentS5.6 で動作確認しています。)

CentOS で何らかの作業をされる方の役に立てれば幸いです。

オープンソースと呼ぶには極めて小規模で、ブログで公開する類のものです。ただ今後、CentOS 以外の別ディストリビュータへの展開も考え、第三者への協力をお願いしやすい SourceForge 様で公開させていただく次第です。

なお、当シェル群のライセンスは MIT ライセンスです。商用利用も可能です。詳細は zip ファイル内の LICENSE.txt をご参考願います。

また、当シェル群の使用で生じたいかなる問題も、著作権者である私は一切責任は負いません。(ブログ等で公開されていた tips 等をまとめたものに著作権を主張するのは躊躇しますが、「使いやすくまとめた点」においては、私のノウハウが生かされているので、ご容赦願います。参考にさせていただいた tips 等は後述しています。)

2 導入方法

2.1 導入前の確認事項

当シェル群では dg、dgi、lu、lus というシェルスクリプト、および la、lat、latr、h、es、ct、psg、td というコマンドエイリアスを提供します。皆様の環境で「名前の衝突」が無いかな、ご確認願います。（仮に衝突があったとしても「当シェル群の名前を変更する」「皆様の環境の名前を変更する」「当シェル群のものでリプレースする」等で対処できます。）

2.2 導入方法

当シェル群は、次の【手順 1】～【手順 6】で導入します。（【手順 5】までは root で行ってください。【手順 6】は当シェル群を使用する OS ユーザごとに行ってください。）

【手順 1】

当シェル群を配置するディレクトリを決定し、作成します。（当資料では/opt/tool/shell とします。）

```
# mkdir -p /opt/tool/shell
```

【手順 2】

zip 解凍後の shell ディレクトリ配下の全ファイルを手順 1 で作成したディレクトリ配下に配置します。（PC から WinSCP 等で転送される場合、必ずバイナリモードで転送してください。）

【手順 3】

配置したファイルに権限を付与します。

```
# chmod 755 /opt/tool/shell/*
```

【手順 4】

当シェル群には「TCP ポートへのリクエストを 16 進数表示する」td というコマンドエイリアスが含まれています。（NIC 識別子は eth0 を前提にしています。）もしサーバに「監視セグメント用 NIC」「サブドメイン用 NIC」「他システムとのインタフェースセグメント用 NIC」等、複数

の NIC が実装されている場合、コマンドエイリアス定義シェル (custom_alias.sh) の td 定義を参考に「td1、td2」等を必要に応じて追加します。

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr **:**:**:**:**:**
:
eth1      Link encap:Ethernet  HWaddr **:**:**:**:**:**
:
↓
# vi /opt/tool/shell/custom_alias.sh
以下省略
```

※td コマンドエイリアスは、IPv4 環境でのみ動作確認しています。IPv6 環境は未確認です。

【手順 5】

ディレクトリ/dev/shm が、次の 3 条件全てを満たしているか確認します。

- tmpfs (OS 再起動で消滅する RAM ディスク) をマウントしていること
- 十分な空き容量があること
- 読み込み権限と書き込み権限が全ユーザーに付与されていること

```
# df -h
Filesystem      サイズ  使用  残り  使用% マウント位置
:
tmpfs           512M    0  512M   0%  /dev/shm
# ls -al /dev/shm
合計 0
drwxrwxrwt  2 root root  40  5月 21 15:43 .
drwxr-xr-x 10 root root 2940  4月 18 19:58 ..
```

※【手順 2】で配置した lu および lus というシェルですが、実行時、一時的に 0 バイトのファイルを最大で 2 個作成します。スクリプトで一時ファイルの削除は行っていますが、トラップ不可能な kill コマンド等で強制終了すると、削除できないケースが生じます。そのため、lu および lus では OS 再起動で消滅する tmpfs に一時ファイルを作成しています。但し、何らかの事情で/dev/shm が使用できない場合、lu および lus のスクリプトファイルを修正してください。(ともに TMP_DIR='/dev/shm'の行を適切な内容に変更してください。)

【手順 6】

当シェル群配置ディレクトリを `PATH` 環境変数に追加すると同時に、コマンドエイリアス定義シェルの `bash` 起動時に自動実行するようにします。**【手順 6】は当シェル群を使用する OS ユーザごとに実施してください。**（`root` 以外の OS ユーザで行う場合、コマンドプロンプトは `#` ではなく `$` となります。）

```
# vi ~/.bashrc  
  
（最終行に以下の 2 行を追加。2 行目の先頭 2 文字は「ドット」と「スペース」です。）  
export PATH=${PATH}:/opt/tool/shell  
. /opt/tool/shell/custom_alias.sh
```

※`PATH` 環境変数の追加方法や、`bash` 起動時の自動実行スクリプト定義方法は他にもあります。皆様の環境で推奨されている方法で行ってください。（厳密には `custom_alias.sh` というシェルは `PATH` の通っているディレクトリに配置する必要はなく、実行権も不要です。気になる方は、`custom_alias.sh` を別ディレクトリに実行権無しで配置する等してください。）

以上で導入は完了です。

3 使用方法

導入が完了すると、次のコマンドやコマンドエイリアスが使用可能となります。

3.1 dg コマンド

概要	任意のディレクトリ配下の全テキストファイルを対象に、キーワード検索 (grep) を行います。サブディレクトリは再帰検索を行います。	
構文	dg [find_root_dir] grep_keyword	
引数	find_root_dir	grep の起点となるディレクトリ。省略時はカレントディレクトリを起点に grep を行います。
	grep_keyword	grep するキーワード。大文字と小文字は区別されます。
使用例	find_root_dir 指定あり	\$ dg /etc/httpd Directory /etc/httpd/conf/httpd.conf:291:<Directory /> /etc/httpd/conf/httpd.conf:294:</Directory> :
	find_root_dir 指定なし	\$ cd /etc/httpd \$ dg Directory ./conf/httpd.conf:291:<Directory /> ./conf/httpd.conf:294:</Directory> :
注意	実行結果をファイルにリダイレクトする場合、必ず「find_root_dir 配下以外のディレクトリ」を出力先として指定してください。 例) カレントディレクトリ配下で access denied を grep する場合 \$ dg "access denied" > ad.log ==>(NG) ではなく \$ dg "access denied" > ../ad.log ==>(OK) としてください。リダイレクト先が grep 対象になる場合があるからです。	
備考	dg は Directory Grep の略です。Find Grep の略である fg は、OS コマンドとして存在するため、dg としました。	

3.2 dgi コマンド

概要	任意のディレクトリ配下の全テキストファイルを対象に、キーワード検索(grep)を行います。サブディレクトリは再帰検索を行います。	
構文	dg [find_root_dir] grep_keyword	
引数	find_root_dir	grep の起点となるディレクトリ。省略時はカレントディレクトリを起点に grep を行います。
	grep_keyword	grep するキーワード。大文字と小文字は区別されません。
使用例	find_root_dir 指定あり	<pre>\$ dgi /etc/httpd directory : (小文字の directory を含む行) /etc/httpd/conf/httpd.conf:291:<Directory /> /etc/httpd/conf/httpd.conf:294:</Directory> :</pre>
	find_root_dir 指定なし	<pre>\$ cd /etc/httpd \$ dgi directory : (小文字の directory を含む行) ./conf/httpd.conf:291:<Directory /> ./conf/httpd.conf:294:</Directory> :</pre>
注意	<p>実行結果をファイルにリダイレクトする場合、必ず「find_root_dir 配下以外のディレクトリ」を出力先として指定してください。</p> <p>例) カレントディレクトリ配下で access denied を grep する場合</p> <pre>\$ dgi "access denied" > ad.log ==>(NG)</pre> <p>ではなく</p> <pre>\$ dgi "access denied" > ../ad.log ==>(OK)</pre> <p>としてください。リダイレクト先が grep 対象になる場合があるからです。</p>	
備考	大文字と小文字を区別しない dg コマンドです。	

3.3 lu コマンド

概要	カレントディレクトリ配下の全ファイルを対象に、指定された期間内に更新されたタイムスタンプを持つファイルをリストアップし、詳細情報を表示します。サブディレクトリは再帰検索を行います。	
構文	lu [from_inf] [to_inf]	
引数	from_inf	期間の開始日時。省略時は本日以降に更新されたタイムスタンプを持つファイルをリストアップします。指定時は 4 桁または 8 桁または 12 桁の形式で指定します。4 桁の場合、月日の MMDD とみなし、年は本日の年が指定されたものとみなします。8 桁の場合、年月日の YYYYMMDD とみなします。12 桁の場合、日時の YYYYMMDDhhmm とみなします。
	to_inf	期間の終了日時。省略時は from_inf 以降に更新されたタイムスタンプを持つファイルをリストアップします。指定時は 4 桁または 8 桁または 12 桁の形式で指定します。4 桁の場合、月日の MMDD とみなし、年は本日の年が指定されたものとみなします。8 桁の場合、年月日の YYYYMMDD とみなします。12 桁の場合、日時の YYYYMMDDhhmm とみなします。
使用例	<p>本日 0 時 0 分 0 秒以降に更新されたファイルのリストアップ</p> <pre>\$ lu -rw-r--r-- 1 root root 1510460 2012/05/21 07:37:46 ./access_log :</pre> <p>本年 5 月 10 日 0 時 0 分 0 秒から 5 月 20 日 23 時 59 分 59 秒に更新されたファイルのリストアップ</p> <pre>\$ lu 0510 0520 -rw-r--r-- 1 root root 466974 2012/05/12 17:16:36 ./ssl_access_log.1 :</pre> <p>2011 年 4 月 1 日 15 時 30 分 0 秒以降に更新されたファイルのリストアップ</p> <pre>\$ lu 201104011530 -rw-r--r-- 1 root root 30996 2011/04/23 10:34:52 ./docs/CHANGES.txt :</pre>	

3.4 lus コマンド

概要	カレントディレクトリ配下の全ファイルを対象に、指定された期間内に更新されたタイムスタンプを持つファイルをリストアップし、ファイル名のみを表示します。サブディレクトリは再帰検索を行います。	
構文	lus [from_inf] [to_inf]	
引数	from_inf	期間の開始日時。省略時は本日以降に更新されたタイムスタンプを持つファイルをリストアップします。指定時は 4 桁または 8 桁または 12 桁の形式で指定します。4 桁の場合、月日の MMDD とみなし、年は本日の年が指定されたものとみなします。8 桁の場合、年月日の YYYYMMDD とみなします。12 桁の場合、日時の YYYYMMDDhhmm とみなします。
	to_inf	期間の終了日時。省略時は from_inf 以降に更新されたタイムスタンプを持つファイルをリストアップします。指定時は 4 桁または 8 桁または 12 桁の形式で指定します。4 桁の場合、月日の MMDD とみなし、年は本日の年が指定されたものとみなします。8 桁の場合、年月日の YYYYMMDD とみなします。12 桁の場合、日時の YYYYMMDDhhmm とみなします。
使用例	<p>本日 0 時 0 分 0 秒以降に更新されたファイルのリストアップ</p> <pre>\$ lus ./access_log :</pre> <p>本年 5 月 10 日 0 時 0 分 0 秒から 5 月 20 日 23 時 59 分 59 秒に更新されたファイルのリストアップ</p> <pre>\$ lus 0510 0520 ./ssl_access_log.1 :</pre> <p>2011 年 4 月 1 日 15 時 30 分 0 秒以降に更新されたファイルのリストアップ</p> <pre>\$ lus 201104011530 ./docs/CHANGES.txt :</pre>	

3.5 コマンドエイリアス

エイリアス	概要
la	ls -al コマンド実行結果のタイムスタンプを秒単位で表示します。i ノード番号を確認したい場合 \$ la -i testfile のように使用してください。
lat	ls -alt コマンド実行結果のタイムスタンプを秒単位で表示します。
latr	ls -altr コマンド実行結果のタイムスタンプを秒単位で表示します。
h	history コマンドのエイリアスです。
es	最後に実行したコマンドの戻り値を表示します。Exit Status の略です。
ct	環境定義ファイルや ini ファイルの有効行のみを表示します。 \$ ct httpd.conf や \$ ct php.ini のように使用してください。Comment Trim の略です。
psg	実行中プロセスリスト (ps -ef コマンド実行結果) から特定キーワードで grep します。 \$ psg httpd のように使用してください。これは \$ ps -ef grep -v grep grep httpd と同じ意味を持ちます。赤字箇所がエイリアスですが、実行中プロセスリストで grep を含むプロセス (grep 実行中プロセスや、引数に grep を含むプロセス等) は表示されません。dg・dgi コマンドの注意で記したミスを犯した場合、psg は使えません。Process Status Grep の略です。
td	TCP ポートへのリクエストを 16 進数表示します。(root ユーザ専用) # td 465 のように使用してください。Tcp Dump の略です。(これは tcpdump コマンドにオプションを加えたエイリアスです。一般ユーザがフルパス指定で tcpdump を実行するとソケット作成権限無しエラーが発生します。)

4 最後に

4.1 参考 Web サイト

当シェル群の作成に際し、以下の Web サイトを参考にさせていただきました。（上段が直接参考にさせていただいた Web サイト、下段がそのトップページです。）

ヤフー様 (dg・dgi)

http://detail.chiebukuro.yahoo.co.jp/qa/question_detail/q1315866533

<http://www.yahoo.co.jp/>

SanRin 舎様 (dg・dgi)

<http://sanrinsha.lolipop.jp/blog/2012/02/%E3%80%90unixlinux%E3%80%91find-xargs-grep.html>

<http://sanrinsha.lolipop.jp/>

アシアル様 (lu・lus)

<http://blog.asial.co.jp/543>

<http://www.asial.co.jp/>

NTT レゾナント様 (la・lat・latr)

<http://oshiete.goo.ne.jp/qa/4503337.html>

<http://www.goo.ne.jp/>

ねっとでべろ様 (ct)

http://yk.tea-nifty.com/netdev/2005/10/post_6839.html

<http://yk.tea-nifty.com/netdev/>

ときどきの雑記帖様 (ct)

<http://www.kt.rim.or.jp/~kbbk/regex/regex.html>

<http://www.kt.rim.or.jp/~kbbk/zakkicho/index.html>

Linux 日和様 (td)

http://linux-biyori.sakura.ne.jp/setting/st_tcpdump.php

<http://linux-biyori.sakura.ne.jp/index.php>

上記以外にも@IT 様、CodeZine 様、日経 BP 様はじめ、技術情報を公開しておられる多くの

Web サイトを参考にさせていただきました。ありがとうございました。

4.2 バージョンについて

オープンソースの場合、最初の公開は 0.1 版等となることが多いようですが、当シェル群は本番環境へのリリースも想定し、1.0 版として公開させていただきました。（「0.1 版を本番環境にリリースするのか？」等の疑念を抱かれる人がおられるかも知れないからです。）

4.3 その他

コマンドエイリアス定義シェル (`custom_alias.sh`) ですが、`vi` と `view` の定義行がコメントになっています。このコメント行を解除すれば、`vi・view` コマンドで `vim` が使用できます。`vim` とは `vi` を拡張したエディタで、優れたカラー表示機能等があり、`vi` と比べると視認性が著しく向上しています。

そのため、例えばログファイルの確認を `view` で実施するようなルールのある運用現場では、`vim` で実施したほうが好ましいです。ところが運用マニュアル等で「`view` でログファイルを開いて `XXX` というキーワードを検索」等と記されていれば、運用マニュアル等も修正せねばなりません。運用マニュアル等を修正せずに `vim` を使用したい場合、`custom_alias.sh` のコメント行を解除してください。なお、`zip` 解凍後の `other` ディレクトリに `.exrc` というファイルがあります。これは `vi` 系エディタの設定ファイルです。なくても構いませんが、あれば便利です。OS ユーザのホームディレクトリ直下に配置すると、有効になります。よろしければお使いください。（デフォルトで「行番号を表示する」「文字列検索で大文字・小文字を区別しない」「対応するカッコを強調表示する」「タブインデントを 4 文字にする」という設定を行っています。）

以上