

cronolog クイックセットアップ

2016/01/09

apache付属の rotatelog だとカレンダーに従ったログローテーションができないので cronolog を導入。そのメモ。

説明

apacheのログファイルローテーションをカレンダーの月単位に行いたいとき `httpd.conf` に

```
CustomLog "|/usr/local/sbin/rotatelog httpd-access.%m.log 2592000" combind
```

と書いてもうまくいかない。

$2592000 = 30 \times 24 \times 60 \times 60 = 30\text{日} \times 24\text{時間} \times 60\text{分} \times 60\text{秒}$ で30日を秒に換算したもの

2592000 はローテーションタイミングで、経過秒数になる。.....Apacheから rotatelog が呼び出されてからの経過時間になるので、例えば1月10日にApacheを起動した場合、実際にログのローテーションが実施されるのは2月9日になる。2月1日にログのローテートを期待していると『あれ?』となってしまう。

カレンダーに従ったローテーション

カレンダーに従ったログローテーションは、cronolog を利用した方がいい。* cronolog ではなく cronolog なので綴りを間違えないように。

```
CustomLog "|/usr/local/sbin/cronolog httpd-access.%m.log" combind %m は 01 ~ 12 になる
```

制限付きだが経過時間による指定も可能

rotatelog のように経過時間による指定も可能だが制限付きになる。

例えば1か月経過でローテーションさせるなら

```
CustomLog "|/usr/local/sbin/cronolog --period=1months httpd-access.%m.log" combind
```

になるし、1週間経過の指定なら

```
CustomLog "|/usr/local/sbin/cronolog --period=1weeks httpd-access.%W.log" combind %W は 00 ~ 53 になる。その年の最初の月曜日の週を00とする(第1週目とする)
```

になる。

単位には seconds, minutes, hours, days, weeks, months が指定できる。

ただし、数値に 1, 2, 3, 4, 5, 6, 10, 15, 20, 30 の指定しかできない。

そのため cronolog では以下のような指定はできない。

```
CustomLog "|/usr/local/sbin/cronolog --period=24hours httpd-access.%d.log"
combind<<<<←"24時間"の指定はできない<<<<"1days"になる<<<<%dは01~31になる。
CustomLog "|/usr/local/sbin/cronolog --period=7days httpd-access.%W.log"
combind<<<<←"7日間"の指定はできない<<<<"1weeks"になる。
CustomLog "|/usr/local/sbin/cronolog --period=12months httpd-access.%Y.log"
combind<<<<←"12か月"の指定はできない<<<<"1years"の指定もできないので実現不可<<<<%Yは1970~2038になる。
```

apache以外からの利用

rotatelogsもそうだが<<<<cronologはapache以外からも利用できる。

```
$ cat testecho.sh
#!/bin/sh
for arg in `seq 1 10`
do
  echo loop $arg
  vmstat
  sleep 10
done

$ sh ./testecho.sh | cronolog testecho.log
$ ll
total 16
drwxr-xr-x  2 k896951  custom - 512  1月 9 14:30 ./
drwxr-xr-x 44 k896951  custom - 2560 1月 9 14:30 ../
-rw-r--r--  1 k896951  custom - 2511 1月 9 14:31 testecho.log
-rw-r--r--  1 k896951  custom - 78  1月 9 14:28 testecho.sh
$ cat testecho.log
loop 1
procs      memory      page      disks      faults      cpu
 r b w      avm    fre flt re  pi  po    fr  sr da0 da1  in  sy  cs us
sy id
0 0 0 4297848 2946872 106  0  0  0  127  9  0  0 236 675 404
0 0 100
loop 2
procs      memory      page      disks      faults      cpu
 r b w      avm    fre flt re  pi  po    fr  sr da0 da1  in  sy  cs us
sy id
0 0 0 4297848 2946868 106  0  0  0  127  9  0  0 236 675 404
0 0 100
loop 3
procs      memory      page      disks      faults      cpu
 r b w      avm    fre flt re  pi  po    fr  sr da0 da1  in  sy  cs us
sy id
0 0 0 4297848 2946828 106  0  0  0  127  9  0  0 236 675 404
0 0 100
loop 4
procs      memory      page      disks      faults      cpu
```

```

 r b w      avm      fre      flt re  pi  po      fr  sr da0 da1  in  sy  cs us
sy id
 0 0 0 4297848 2946824 106 0 0 0 127 9 0 0 236 675 404
0 0 100
loop 5
procs      memory      page      disks      faults      cpu
 r b w      avm      fre      flt re  pi  po      fr  sr da0 da1  in  sy  cs us
sy id
 0 0 0 4297848 2946820 106 0 0 0 127 9 0 0 236 675 404
0 0 100
loop 6
procs      memory      page      disks      faults      cpu
 r b w      avm      fre      flt re  pi  po      fr  sr da0 da1  in  sy  cs us
sy id
 0 0 0 4297848 2946816 106 0 0 0 127 9 0 0 236 675 404
0 0 100
loop 7
procs      memory      page      disks      faults      cpu
 r b w      avm      fre      flt re  pi  po      fr  sr da0 da1  in  sy  cs us
sy id
 0 0 0 4297848 2946780 106 0 0 0 127 9 0 0 236 675 404
0 0 100
loop 8
procs      memory      page      disks      faults      cpu
 r b w      avm      fre      flt re  pi  po      fr  sr da0 da1  in  sy  cs us
sy id
 0 0 0 4297848 2946776 106 0 0 0 127 9 0 0 236 675 404
0 0 100
loop 9
procs      memory      page      disks      faults      cpu
 r b w      avm      fre      flt re  pi  po      fr  sr da0 da1  in  sy  cs us
sy id
 0 0 0 4297848 2946772 106 0 0 0 127 9 0 0 236 675 404
0 0 100
loop 10
procs      memory      page      disks      faults      cpu
 r b w      avm      fre      flt re  pi  po      fr  sr da0 da1  in  sy  cs us
sy id
 0 0 0 4297848 2946768 106 0 0 0 127 9 0 0 236 675 404
0 0 100
$

```

10秒ごとにローテーションするならこんな感じ。

```

$ sh ./testecho.sh | cronolog --period=10seconds testecho.%S.log
$ ll
total 44
drwxr-xr-x  2 k896951  custom - 512  1月 9 14:38 ./
drwxr-xr-x 44 k896951  custom - 2560 1月 9 14:30 ../
-rw-r--r--  1 k896951  custom - 502  1月 9 14:38 testecho.05.log
-rw-r--r--  1 k896951  custom - 503  1月 9 14:38 testecho.15.log

```

```

-rw-r--r-- 1 k896951 custom - 251 1月 9 14:37 testecho.25.log
-rw-r--r-- 1 k896951 custom - 251 1月 9 14:37 testecho.35.log
-rw-r--r-- 1 k896951 custom - 502 1月 9 14:37 testecho.45.log
-rw-r--r-- 1 k896951 custom - 502 1月 9 14:38 testecho.55.log
-rw-r--r-- 1 k896951 custom - 2511 1月 9 14:31 testecho.log
-rw-r--r-- 1 k896951 custom - 78 1月 9 14:28 testecho.sh
$ cat testecho.05.log
loop 3
procs          memory          page          disks          faults          cpu
r b w          avm      fre   flt re  pi  po    fr  sr da0 da1    in  sy  cs us
sy id
0 0 0 4297848 2946732 106  0  0  0 127  9  0  0 236 675 405
0 0 100
loop 9
procs          memory          page          disks          faults          cpu
r b w          avm      fre   flt re  pi  po    fr  sr da0 da1    in  sy  cs us
sy id
0 0 0 4297848 2946696 106  0  0  0 127  9  0  0 236 675 405
0 0 100
$ cat testecho.15.log
loop 4
procs          memory          page          disks          faults          cpu
r b w          avm      fre   flt re  pi  po    fr  sr da0 da1    in  sy  cs us
sy id
0 0 0 4297848 2946724 106  0  0  0 127  9  0  0 236 675 405
0 0 100
loop 10
procs          memory          page          disks          faults          cpu
r b w          avm      fre   flt re  pi  po    fr  sr da0 da1    in  sy  cs us
sy id
0 0 0 4297848 2946692 106  0  0  0 127  9  0  0 236 675 405
0 0 100
$ cat testecho.25.log
loop 5
procs          memory          page          disks          faults          cpu
r b w          avm      fre   flt re  pi  po    fr  sr da0 da1    in  sy  cs us
sy id
0 0 0 4297848 2946716 106  0  0  0 127  9  0  0 236 675 405
0 0 100
$ cat testecho.35.log
loop 6
procs          memory          page          disks          faults          cpu
r b w          avm      fre   flt re  pi  po    fr  sr da0 da1    in  sy  cs us
sy id
0 0 0 4297848 2946708 106  0  0  0 127  9  0  0 236 675 405
0 0 100
$ cat testecho.45.log
loop 1
procs          memory          page          disks          faults          cpu
r b w          avm      fre   flt re  pi  po    fr  sr da0 da1    in  sy  cs us
sy id

```

```

0 0 0 4297848 2946748 106 0 0 0 127 9 0 0 236 675 404
0 0 100
loop 7
procs      memory      page
r b w      avm    fre   flt re  pi  po      fr  sr da0 da1    in  sy  cs us
sy id
0 0 0 4297848 2946704 106 0 0 0 127 9 0 0 236 675 405
0 0 100
$ cat testecho.55.log
loop 2
procs      memory      page
r b w      avm    fre   flt re  pi  po      fr  sr da0 da1    in  sy  cs us
sy id
0 0 0 4297848 2946740 106 0 0 0 127 9 0 0 236 675 405
0 0 100
loop 8
procs      memory      page
r b w      avm    fre   flt re  pi  po      fr  sr da0 da1    in  sy  cs us
sy id
0 0 0 4297848 2946700 106 0 0 0 127 9 0 0 236 675 405
0 0 100
$

```

%Sの部分は実行のタイミングで変わる。この例だと、

ログファイル	ログ記録内容
testecho.45.log	loop 1□loop 7
testecho.55.log	loop 2□loop 8
testecho.05.log	loop 3□loop 9
testecho.15.log	loop 4□loop 10
testecho.25.log	loop 5
testecho.35.log	loop 6

testecho.sh 起動時のタイマーが45秒だったことがわかる。また10秒おきに記録先のログファイルが切り替わっていることも確認できる。

手順

ports の sysutils/cronolog を make; make install する。

```

root@amanda:/root # cd /usr/ports/sysutils/cronolog
root@amanda:/usr/ports/sysutils/cronolog # make clean
====> Cleaning for cronolog-1.6.2_5
root@amanda:/usr/ports/sysutils/cronolog # make
====> License GPLv2 accepted by the user
====> Found saved configuration for cronolog-1.6.2_5
====> cronolog-1.6.2_5 depends on file: /usr/local/sbin/pkg - found
====> Fetching all distfiles required by cronolog-1.6.2_5 for building
====> Extracting for cronolog-1.6.2_5

```

```
-- 中略 --

====>  cronolog-1.6.2_5 depends on executable: indexinfo - found
====>  Generating temporary packing list
Making install in lib
Making install in src
/bin/sh ../mkinstalldirs
/usr/ports/sysutils/cronolog/work/stage/usr/local/sbin
  install -s -m 555  cronolog
/usr/ports/sysutils/cronolog/work/stage/usr/local/sbin/cronolog
/bin/sh ../mkinstalldirs
/usr/ports/sysutils/cronolog/work/stage/usr/local/sbin
  install -m 555 cronosplit
/usr/ports/sysutils/cronolog/work/stage/usr/local/sbin/cronosplit
Making install in doc
/bin/sh ../mkinstalldirs
/usr/ports/sysutils/cronolog/work/stage/usr/local/info/
  install -m 0644 ./cronolog.info
/usr/ports/sysutils/cronolog/work/stage/usr/local/info//cronolog.info
  install-info --info-
dir=/usr/ports/sysutils/cronolog/work/stage/usr/local/info/
/usr/ports/sysutils/cronolog/work/stage/usr/local/info//cronolog.info
/usr/bin/make install-man1
/bin/sh ../mkinstalldirs
/usr/ports/sysutils/cronolog/work/stage/usr/local/man/man1
  install -m 0644 ./cronolog.1m
/usr/ports/sysutils/cronolog/work/stage/usr/local/man/man1/cronolog.1m
  install -m 0644 ./cronosplit.1m
/usr/ports/sysutils/cronolog/work/stage/usr/local/man/man1/cronosplit.1m
Making install in testsuite
====> Compressing man pages (compress-man)
root@amanda:/usr/ports/sysutils/cronolog # make install
====>  Installing for cronolog-1.6.2_5
====>  cronolog-1.6.2_5 depends on executable: indexinfo - found
====>  Checking if cronolog already installed
====>  Registering installation for cronolog-1.6.2_5
Installing cronolog-1.6.2_5...
root@amanda:/usr/ports/sysutils/cronolog #
```

[技術資料](#), [FreeBSD](#), [apache](#), [口グ](#), [cronolog](#)

From:

<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:

<https://wiki.hgotoh.jp/documents/quick/quick-0014>

Last update: **2023/04/14 02:32**

