NAME

fileprune − prune a file set according to a given age distribution

SYNOPSIS

fileprune [**−n**|**−N**|**−p**] [**−c** *count*|**−s** *size*[**k**|**m**|**g**|**t**]|**−a** *age*[**w**|**m**|**y**]] [**−e** *base*|**−g** *standard deviation*|**−f**] [**−t a**|**m**|**c**] [**−FKv**] *file ...*

fileprune -d −n|**−N [−c** *count*|**−a** *age***[w**|**m**|**y]] [−e** *base*|**−g** *standard deviation*|**−f] [−FKv]** *date ...*

DESCRIPTION

Fileprune will delete files from the specified set targeting a given distribution of the files within time as well as size, number, and age constraints. Its main purpose is to keep a set of daily-created backup files in manageable size, while still providing reasonable access to older versions. Specifying a size, file number, or age constraint will simply remove files starting from the oldest, until the constraint is met. The distribution specification (exponential, Gaussian (normal), or Fibonacci) provides finer control of the files to delete, allowing the retention of recent copies and the increasingly aggressive pruning of the older files. The retention schedule specifies the age intervals for which files will be retained. As an example, an exponential retention schedule for 10 files with a base of 2 will be

1 2 4 8 16 32 64 128 256 512 1024

The above schedule specifies that for the interval of 65 to 128 days there should be (at least) one retained file (unless constraints and options override this setting). Retention schedules are always calculated and evaluated in integer days. By default *fileprune* will keep the oldest file within each day interval allowing files to migrate from one interval to the next as time goes by. It may also keep additional files, if the complete file set satisfies the specified constraint. The algorithm used for pruning does not assume that the files are uniformly distributed; *fileprune* will successfully prune file collections stored at irregular intervals.

OPTIONS

−n Do not delete files; only print file names that would be deleted.

−N Do not delete files; only print file names that would be retained.

−p Do not process files. Print the specified schedule for *count* elements.

−c *count*
Keep *count* files.

−s *size* Keep files totaling *size* bytes. The *size* argument can be followed by a **k**, **m**, **g**, or **t** uppercase or lowercase suffix to express quantities from kilobytes to terabytes.

−a *age* Keep files up to the specified *age*. The *age* argument can be followed by a **w**, **m**, or **y** suffix to specify weeks, months, or years.

−e *base* Use an exponential distribution of the specified *base b* for pruning. Each successive interval *n* will end at $b^{n.}$ As an example, a base of 2 will retain 10 files in a period of 1024 days. To determine the exponent for keeping *n* files in a period of *d* days use the formula $exponent = e^{\ln\frac{d}{n}}$

−g *sd* Use a Gaussian (normal) distribution with the given *standard deviation* for the pruning schedule. The height of the curve with a standard deviation of σ is given by the formula $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}}$ All intervals from *a* to *b* are calculated to have the same $\int_a^b f(x)dx$ The standard deviation is specified in day units; as a rule of a thumb the oldest file retained will have an age of twice the standard deviation.

−f Use a Fibonacci distribution for the pruning schedule. The Fibonacci sequence starts with 0, 1, and each subsequent term is the sum of the two previous ones.

−t a|fP|c
For determining a file's age use its access, modification, or creation time. By default the modification time is used.

−F Force file pruning even if the size or count constraint has not been exceeded.

−K Keep files scheduled in each pruning interval, even if the size or count constraint has been exceeded.

−v Operate in verbose mode, printing the name of each deleted file. Specifying this option a second time will print additional debugging information.

−d Use a list of ISO dates rather than files as an argument of the pruning schedule. Each date argument must be of the form *YYYY-MM-DD* [*hh*[:*mm*[:*ss*]]]. This option must be used with one of the **−N** or **−n** options, and cannot be combined with the **−t** or **−s** options.

EXAMPLE

```
ssh remotehost tar cf - /datafiles >backup/‘date +’%Y%m%d’‘
fileprune -e 2 backup/*
```

Backup *remotehost*, storing the result in a file named with today's timestamp (e.g. 20021219). Prune the files in the backup directory so that each retained file's age will be double that of its immediately younger neighbor.

```
fileprune -N -d -e 1.2 -c 40 *
```

Keep at most 40 files. This particular distribution will result in daily copies for the first fortnight, at least weekly for the next month, and almost monthly for the first year.

```
fileprune -g 365 -c 30 *
```

Keep at most 30 files with their ages following a Gaussian (normal) distribution with a standard deviation of one year.

```
fileprune -e 2 -s 5G *
```

Prune the specified files following an exponential schedule so that no more than 5GB are occupied. More than one file may be left in an interval, if the size constraint is met. Alternatively, some old intervals may be emptied in order to satisfy the size constraint.

```
fileprune -F -e 2 -s 5G *
```

As above, but leave no more than one file in each scheduled interval.

```
fileprune -K -e 2 -s 5G *
```

As in the first example of the %g-constrained series, but leave exactly one file in each interval, even if this will violate the size constraint.

```
fileprune -a 1m -f
```

Delete all files older than one month use; use a Fibonacci distribution for pruning the remaining ones.

```
SNAPSHOTS=/tmp/snapshots.$$
ec2-describe-snapshots --filter status=completed |
awk ’$1 == "SNAPSHOT" {print $2, substr($5, 1, 10)}’ |
sort -k2 >$SNAPSHOTS
fileprune -n -d -e 1.2 -c 40 ‘awk ’{print $2}’ $SNAPSHOTS‘ |
sort |
join -1 1 -2 2 -o 2.1 - $SNAPSHOTS |
xargs -n 1 ec2-delete-snapshot
rm -f $SNAPSHOTS
```

Prune AWS-hosted daily snapshots to leave 40.

SEE ALSO

newsyslog(8)

AUTHOR

BUGS

The Gaussian (normal) distribution is calculated by trying successive increments of the normal function's distribution function. If the file number or count is large compared to the specified standard deviation, the calculation may take an exceedingly long time. To get results in a reasonable time, day increments are bounded at 10 times the increment of the previous interval and a total age of 100 years. It is advisable to first calculate and print the pruning schedule with a command like

```
fileprune -g 100 -p -c 20
```

to ensure that the schedule can be calculated.