

Automatic file backups with `cron` and `tar`

`mschul@mcs.mu.edu`

October 6, 2007

What is `cron`

`cron` is a simple utility which will automatically run a program at a given time.

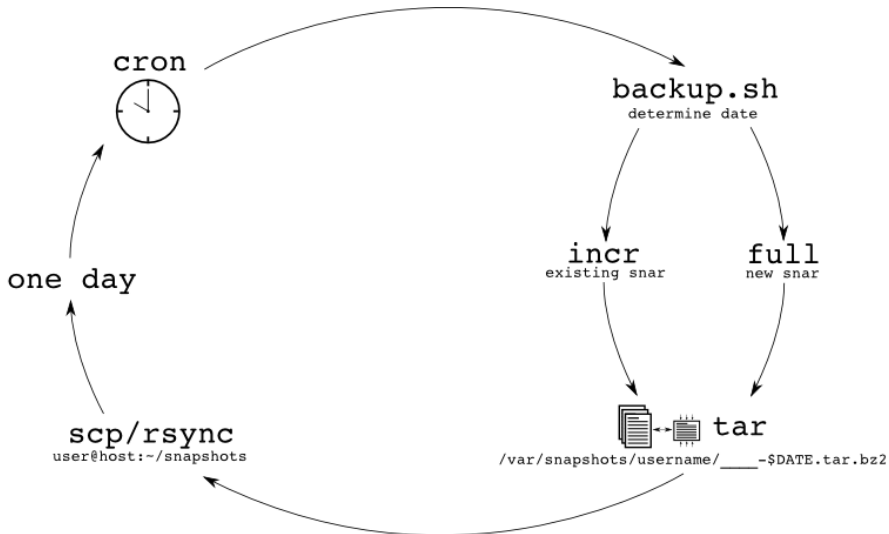
- System wide daemon
- User level crontabs
- `crontab -l` lists current crontab

What is `tar`

`tar` (short for "tape archiver"), archives files/directories into one.

- Easily put an entire directory into one file.
- Command line options for compression.
- `tar -t` lists file archived in tarball.

Overview of the process



Writing the script

Select a command interpreter:

- `#!/bin/sh`

Setup some simple variables:

- `SNAPDIR=/var/snapshots/$USER`
- `RMT_DIR="user@hostname:~/snapshots"`

Variable set by executing shell commands:

- `DATE=$(date +%Y%m%d)`
- `TAR=$(which tar)`

Using a combination of existing variables, shell commands, and pipes:

- `LAST_FULL=$(stat -f "%Dc %Sc" -t "%Y%m%d" ${SNAPDIR}/full-*.tar.gz
2> /dev/null | sort -n | tail -n1)`
- `LAST_TS=$(echo ${LAST_FULL} | awk '{ print $1}')`

`man tar`

`man tar` shows us a lot of options to use with the `tar` command, of most interest to us are:

- `z` use the gzip compression algorithm

`man tar`

`man tar` shows us a lot of options to use with the `tar` command, of most interest to us are:

- `z` use the gzip compression algorithm
- `c` create a new archive

`man tar`

`man tar` shows us a lot of options to use with the `tar` command, of most interest to us are:

- `z` use the gzip compression algorithm
- `c` create a new archive
- `f` using the following filename ...

man tar shows us a lot of options to use with the tar command, of most interest to us are:

- z use the gzip compression algorithm
- c create a new archive
- f using the following filename ...
- --exclude-from do not use files listed in ...

man tar shows us a lot of options to use with the tar command, of most interest to us are:

- z use the gzip compression algorithm
- c create a new archive
- f using the following filename ...
- --exclude-from do not use files listed in ...
- --listed-incremental use file in ... for GNU-format incremental backup

Using `tar` in the script

```
function incr {
    ${TAR} czf ${SNAPDIR}/incr-${DATE}.tar.gz \
        --exclude-from $HOME/.snap-exclude \
        --listed-incremental=${SNAPDIR}/$USER-${LAST_DATE}.snar \
        $HOME > /dev/null 2> /dev/null
    ${CHMOD} go-rwx ${SNAPDIR}/incr-${DATE}.tar.gz
}
```

Creating the offsite backup

```

${RMT_CMD} ${RMT_OPTIONS} ${SNAPDIR}/full-${DATE}.tar.gz \
  ${SNAPDIR}/${USER}-${DATE}.snar \
  ${RMT_DIR}

```

I use `scp` to push the most current data to an offsite system.
`rsync` would also work with a few adjustments.

Miscellaneous ends

- cleanup—remove files older than one month, if needed
- remote cleanup—scp doesn't allow removing file, rsync does though
- transferring files—automatic, no password, no passphrase, insecure?

Credits

I have placed the source code and a copy of this presentation online

- <http://mulug.mscs.mu.edu/~mjschultz/backup/>

This presentation was created using the following open source tools:

- Inkscape: <http://www.inkscape.org/>
- \LaTeX : <http://latex-project.org/>
- Vim: <http://www.vim.org/>

Special thanks to the Marquette University Linux Users Group for helping correct errors in this presentation.