

**Ad
System
I
n
I**

Backups

**t
r
a
t
i
o**

U n IX

Backing Up And Restoring Files

⌘ a disaster recovery mechanism

- ⌘ beware of murphy's law!
- ⌘ some users use backups as a version control mechanism
 - ⌘ *probably worthwhile discouraging these users*
 - and making some other facility available to them...

⌘ one of the priority tasks for any system administrator

- ⌘ often, a large proportion of the wealth of a company is tied up on it's computers
 - ⌘ *the administrator has to be responsible for that wealth*
 - responsibility is often specified by the employment contract
- ⌘ often delegated to a specific group of administrators
 - ⌘ *big sites often decide to have a dedicated backup pseudo user equivalent to root (UID = 0) but whose activities can be tracked separately*

⌘ treated like a chore by many people

- ⌘ so, often not done properly
- ⌘ should be undertaken holistically
 - ⌘ *a mixture of automation/policy/procedure is needed for effective operation*

Never Trust The Media!

⌘ in this case, the physical tapes, disks, CDs, etc.

☒ *all media has a limited lifetime*

☒ *in the best of circumstances*

☒ *in a hot, humid climate...?!*

⌘ **make sure that:**

☒ all files are backed up on multiple volumes

☒ *multiple media types are also a good idea*

☒ all backups are tested

☒ *no point in having a backup on a bad tape...!*

☒ there exists a place for offsite storage

☒ you have an 'emergency' plan

☒ *for use when the fire alarm goes*

Incremental Backup Strategies

⌘ may be too difficult to do full backups all the time

☒ store only 'recently' changed files

☒ backup strategy defined by

☒ ease

- should be a "no-brainer" to do
- may be difficult if the data to be backed up is never quiescent
 - need to adopt very careful techniques

☒ capacity of backup volumes

- no point in devising a complicated incremental strategy if the volumes are large enough to hold everything

☒ frequency of change of data

- no point in repeatedly doing full dumps of static filesystems

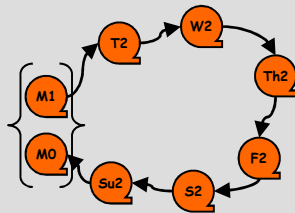
⌘ typically divide backups into sets (levels)

☒ typically define 10 backup levels. A level 0 backup saves all files. Subsequent backups store those files modified since the last backup at a lesser level.

☒ may not be supported by the program but the idea is valid

☒ (see p.470, Frisch)

☒ key idea of all schemes is to ensure that data appears on more than one volume to guard against corruption



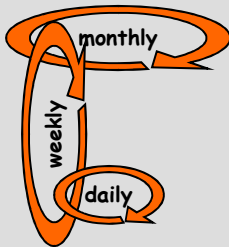
Another Strategy

⌘ suggested by the dump manual page

☒ in the event of a catastrophic disk event, the time required to restore all the necessary backup tapes or files to disk can be kept to a minimum by staggering the incremental dumps. An efficient method follows:

☒ *always start with a level 0 backup, for example:*

```
/sbin/dump 0uf /dev/tape /home
```



☒ *this should be done at set intervals, say once a month or once every two months, and on a set of fresh tapes that is saved forever.*

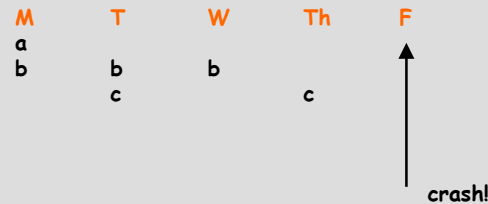
☒ *after the level 0 dump, dumps of active file systems are taken on a daily basis, using a modified Tower of Hanoi algorithm, with this sequence of dump levels: 3 2 5 4 7 6 9 8 9 9 ...*

☒ *for the daily dumps, it should be possible to use a fixed number of tapes for each day, used on a weekly basis. Each week, a level 1 dump is taken, and the daily Hanoi sequence repeats beginning with 3. For weekly dumps, another fixed set of tapes per dumped file system is used, also on a cyclical basis.*

☒ *after several months or so, the daily and weekly tapes should get rotated out of the dump cycle and fresh tapes brought in*

Incremental Restoration

- ⌘ always involves going through volumes from *oldest* to *youngest*



- ⌘ usually best to ensure that restored files don't overwrite current files
 - ☒ always good policy!
 - ☒ user may need to merge the different versions, not replace them
 - ☒ may be difficult with some tools

System-Level Backups

⌘ two main programs

⌘ dump

⌘ *backup on a per-filesystem basis*

```
# dump 0uf /dev/tape /dev/hda
```

⌘ *supports the idea of levels*

- maintains a per-level record in /etc/dumpdates (if it exists)

⌘ *volume spanning solves problem of large backup size*

⌘ *should only be used on an unmounted filesystem, but...*

⌘ */etc/fstab contains a dump frequency field. Usually ignored!*

⌘ *rdump variant can dump to a remote tape drive across a network*

⌘ restore

⌘ *complement to dump*

⌘ *restore either whole filesystem or individual files*

- can be browsed interactively
 - best way to restore a small number of files/directories

```
# restore if /dev/tape
```

- (see p.487, Frisch)

⌘ *can be used to verify the contents of a dump*

- (see p.488, Frisch)

User-Level Backups

⌘ several programs available

- ☒ cpio, tar (probably most popular), pax
- ☒ simple to use

```
% tar -c -v -f myfiles.tar ./myfiles
% tar -c -f - . | (cd todir; tar -x -v -f -B -p -)
```

- ☒ can also dump across a network

☒ *in conjunction with rsh*

```
% tar -c -f -b - 20 ./macau\ unix\ course/ | rsh redhat 'dd of=/dev/tape obs=20b'
```

- ☒ main problems:

☒ *difficult to use to span volumes*

- difficult to use for the non-technical user, full stop!

☒ *getting the files quiescent*

- users often don't understand this...

☒ *selective restoration difficult*

☒ *restoring to a different location can be tricky*

- should always backup *relative* to current directory (esp. for tar)

☒ *doesn't directly support the ideas of backup levels*

- (see p.480, Frisch)

☒ *these programs actually vary wildly between systems*

- so getting a portable backup is not automatic
 - this is a general rule of backups...

Filesystem Usage

df

☞ displays the amount of disk space available on the filesystems

```
# df -k
Filesystem      1024-blocks  Used Available Capacity    Mounted on
/dev/hda1         49718    30067     17084      64%      /
/dev/sdb1        100118    16984     77964      18%     /home
/dev/sdb5        411371   269159    120965      69%     /usr
/dev/hda6        134452     8654    118855       7%     /var
/dev/sda1         19809        13     18773       0%  /scratch1
/dev/sda2         19825        165     18636       1%     /tmp
```

du

☞ displays the amount of disk space used by each argument and for each subdirectory of directory arguments

```
# du -s -k ~bob
15541    /home/bob
```