MAKING ARCHIVES WORK.

In the chapter reasonably entitled *ARCHIVING*, we introduced the idea of the archive system as an extension of the file system onto – typically – off-line media. Now we offer some material on implementing archive systems. There are two components to the task. One is the system which resides in the computer and deals with requests for archive operations as they are received, and the second is the archive processing run, when information is exchanged between computer and archive medium.

The computer-resident component deals with transactions using information from a resident archive directory, and an important task in designing an archive system is to determine just what information must be saved in the resident directory in order to support the required range of operations. For an automatic archive system, the requests are for archive directory services, including requests for information about archived files and such operations on files as can be executed by operations on the directories or attributes, and for the retrieval of archived files. Some of these – details depend on the system – can be executed immediately using information kept in memory (though it might be necessary to carry through directory changes to archived directories when convenient), while others cannot be implemented until the archive medium is present. Much the same comments apply to a discretionary system, but in this case it is also necessary to administer a computer-resident repository of files to be archived.

The archive run is essentially an update operation on the archive material. In most cases, it is a tape-to-tape (that's old-tape-to-new-tape) copy of the archive, with necessary modifications handled as the appropriate point in the tape-to-tape stream passes through the computer. To make this work effectively, quite a lot of planning is necessary; there is a good description of the important points in the IBM4341 example which follows.

EXAMPLES.

The University's Computer Centre used to operate two archiving systems, one for an IBM4341 computer and one for a Vax cluster. The IBM4341 Archiver was a purely discretionary system, while the Vax Archiver provided automatic file saving, with discretionary retrieval. Though neither system is now in use, they are good examples, as they are comparatively straightforward and uncluttered specimens.

THE VAX ARCHIVER[IMP30].

The Archive system automatically maintains magnetic tape copies of the latest version of most of your files. Each time the Daily_Archive job is run (each working day at 7.30 a.m. at Auckland University), it scans directories specified by the system manager and locates any files which you have created or modified since the previous run, and makes two copies of each. At any time you may

- Inquire : to find out what files you have in the archive;
- Dearchive : request that some specified archived files be recreated on disk;
- Delete some specified files from the Archive.

Some files cannot be archived, viz:

- Files greater than the size (in blocks) specified in the logical name ARKMAXFILE;

- Files which do not have System:Read Access. (This includes files in directories which do not have System:Read access),

- Files which are already open for write-access at the time of the Archive run;

- Files which are deaccess-locked.

The archive system will never delete your files from disk, nor will files be deleted from the Archive unless you explicitly request them to be, or unless you request your usercode to be cancelled. If, by performing an Inquire, you know that a file is archived, you may safely delete it from disk to reduce your on-line holdings. A file is deemed to be yours if its owner matches your UIC.

The Dearchive runs as a batch job. If you issue the Dearchive command interactively, a batch job will be submitted for you. If you use the command from a batch job, it will be performed in-line. When the Dearchive has been completed, the log file DEARCHIVE.LOG in your current directory should be checked for any errors. Do not delete the file produced in step 2, or create a higher version of it, until the Dearchive is completed.

If your Dearchive job fails unexpectedly, check that

- you have not exceeded your disk quota

- you have write access to your current directory

- the file specified in the Dearchive command exists

- all directories that you wish to dearchive files into exist. The Archive cannot re-create these for you without risking violating system security.

If any file cannot be sucessfully dearchived, the job aborts. If you need to restart the Dearchive, edit the file of names to remove any names that were sucessfully retrieved, and resubmit the job.

For your benefit, files are never deleted from the Archive unless you request them to be. It is however in everyone's interest for you to periodically delete unwanted archived files. The syntax of the Delete command is:

$ ARCHIVE DELETE filename [NOCONFIRM]

Note that the filename specified is the one containing the list of files to be deleted. If the NOCONFIRM option is specified, the files will then be deleted. If it is not specified, each filename will be displayed, and you will be asked to reply "ok" if the file is to be deleted.

It is possible that you will be told that the Archive database is temporarily locked against update access, and that you may wait or break out with Ctrl/C and try later. This can happen if the system operators are running an Archive job which requires excluslve access to the database.

NOTES.

Observe the reference to "system:read access". This is another form of protection implemented as a protection code.

The "disk quota" is a limit on the amount of disc space you may use. We discuss it in the later section on *IMPLEMENTATION*.

UIC stands for User Identification Code (we think).

THE IBM4341 ARCHIVER[IMP31].

Every user of CMS on the 4341 has a 'Virtual Machine' (VM) which amongst other things provides memory to work in, and disk(s) to store files on. A Virtual Machine's disk space is permanently allocated to it, so when you create files on your own disks they stay there until you ERASE them. If you run out of room on your disks you need somewhere else to keep your less-frequently-used files; this is what Archiver (i.e. the Archive System) provides.

To copy a file into archive, enter

```
ARCHIVE SAVE fileid
```

where fileid is the file's CMS name, e.g, TESTPROG FORTRAN, SAMPLE DATA B, ... When Archiver tells you the file has been archived you can ERASE your copy. To get file back from archive, enter

```
ARCHIVE RECALL fileid DISK
```

Archiver will copy the file back to your disk for you.

In the above SAVE example Archiver wasn't told how long you wanted to keep the file, so it would expire (i.e. be automatically expunged) after two weeks. To say how long the file should be kept, specify its LIFETIME, e,g.

```
ARCHIVE SAVE fileid LIFETIME 6 MONTHS
```

tells Archiver that the file should expire 6 months from today.

2.l OVERVIEW

'Archiver' is a virtual machine with a large amount of disk space (400 Mbytes at the University of Auckland) which it uses as short-term storage for archived files. For longer-term storage Archiver has a collection of magnetic tapes. These tapes are organised into archive 'levels', with level 1 tapes containing the most-recently archived files, level 2 containing older files, and so on. Files enter the archive by being copied onto archive disk; from there they can be copied back immediately when a user recalls them. When archive disk is nearly full, the oldest files on it are 'raked' out to level 1 tape, i,e. copied to tape and erased from disk. If they are not recalled from level l they will eventually be raked to level 2 tape.

As well as the archived files, archiver maintains files of its own, the most important of which is "Catalog". (Note on

terminology: "Catalog" is the proper name of the catalogue – filenames can't be longer than 8 characters.) Catalog is a complete catalogue of all the archived files; it is described in the next section.

The Archive Virtual Machine is normally logged in as a disconnected service VM and is running ARCHEUS, the Archiving program. ARCHEUS just sits waiting for a user request. When it receives one it performs it, then sits back to wait for the next.

Users enter archive commands by invoking the ARCHIVE EXEC. This executes the ARCHIVE program which checks the syntax of the command, then requests ARCHEUS to carry it out. When ARCHIVE has finished it displays a message indicating whether or not your command has been carried out, then it returns to CMS. Its return code is zero if all was well, and non-zero otherwise. The ARCHIVE return codes are set out in a later section.

Requests are passed between your VM and the Archive VM through VM's Inter-User Communication Vehicle, IUCV. This is a part of the system software which makes it possible to send messages between cooperating Virtual Machines. As well as requests for action, ARCHIVE and ARCHEUS use IUCV to move files. For example when you enter a SAVE command ARCHIVE reads the file. from your disk and sends it to ARCHEUS in 2048-byte sections. In earlier stages of Archiver's development other methods of passing requests and files (such as sending virtual card files and copying files between LINKed disks) were used; these are much less effective than IUCV. Note that you shouldn't disturb ARCHIVE while it is executing; if you did (e.g. by entering an HX command) you would probably leave a collection of unanswered IUCV messages from ARCHEUS, which would confuse ARCHIVE next time you ran it. To clear this situation you would have to log off then log on again .

2.2 THE ARCHIVE CATALOGUE

"Catalog" (the Archive catalogue) contains one entry for every file currently in the archive. Each file's Catalog entry contains details such as

- Its filename, filetype and filemode. Note, however, that only its filemode number is remembered; its filemode letter (i.e. the disk it came from) is not.
- The time and date when it was archived.
- Its expiry date. It will be expunged from archive after this date; see the 'Expired Files' section.
- Its access control list. This specifies who can use it; see the 'Sharing Files' section.
- A user's comment of up to 40 characters, which compensates somewhat for CMS's 8-character filenames; see the 'User Comment Syntax' section.
- The location of each copy of the file. Archiver automatically keeps two copies of each file, in different places (i.e. one copy on disk and one on tape, or two tape copies on different reels of tape).

2.3 MANAGING ARCHIVED FILES

Once a file has been put into archive by the archive save command it is Archiver's responsibility to look after it until it is expunged (or it expires). As mentioned in the previous section Archiver maintains two copies of every file, in different places. This means that Archiver must decide where to put each copy, and keep track of the copies as they move about when archive tapes are copied. To this end Archiver is assisted by the 'Helper' Virtual Machine.

Helper contains a catalogue of the archive tapes (the 'tape information' file), and the Helper program. Helper is run at least once each day by the computer operators to perform housekeeping tasks such as

- Making backup copies of newly saved files.
- Clearing archive disk space by 'raking' files to tape.
- Retrieving files from tape back to disk.
- Checking for files which are about to expire.

When a file is saved, Archiver copies it onto the Archive disk and enters it in the archive catalogue. At that stage there is only one copy of it in archive. The next time Helper carries out its Backup task, it will write copies of all such fi1es to a 'level 1' tape and record their whereabouts (tape name and position on tape) in the catalogue. From then on there are two copies of the file in archive; in the unlikely event that Archiver is unable to read the first, it will try to recover by reading the second. Helper makes sure that the two copies of a file are never on the same tape so that it can survive the loss of a whole reel of tape.

As time goes by the archive disks fill up with user files, until Helper decides it is time to rake some disk files to a level 1 tape. Every file's Catalog entry contains the date it was last accessed; Helper uses this 'last-use' date to find the 'oldest' files on disk. These files are copied to tape, they are erased from the archive disk, and their catalog entries updated accordingly. Both archive's copies of such files are now on level 1 tape.

Details of the archive tapes are kept in Helper's 'tape information' file. The tapes are grouped in 'levels'; level 1 tapes contain the 'most recent' files, level 2 tape files are older, and so on. When there isn't much space left for files on the level 1 tapes, Helper will rake some files from level 1 to level 2.

Of the tapes in each level one is always unused; this is the level's 'free' tape. When Helper wants to copy files to a given level it searches the tape information file to select that level's least-full tape; this is the 'next' tape. It scans the archive catalogue making a directory of all the files which are on the 'next' tape then copies them from the 'next' to the 'free' tape, followed by the new files being moved to the level. Before a 'new file' is copied Helper

checks to see that it isn't already on the 'next' tape; if it is there Helper leaves it where it is. This check ensures that a file's second copy is never on the same reel as its first copy. When a file is expunged from archive its Catalog entry is deleted, hence it won't be in Helper's tape directory and it won't be copied to 'next'; it will disappear from archive. Once the copy operation is complete (and all details safely entered in the catalogue) Helper updates the tape information file to make the old 'next' tape 'free'.

3.1 RECALLING FILES

When a user recalls a file, Archiver looks it up in the archive catalogue. If there is a copy on disk it is sent back to the user and its 'last-use' date changed in the catalogue. It thus becomes one of the 'most-recently used' files, hence it will be some time before Helper decides to rake it to tape.

If the file isn't on disk Archiver can't send it back – instead it marks its Catalog entry as 'to be recalled', and sends a "Requested file on tape" message back instead.

Once each working day (usually early in the morning) the computer operators run Helper's 'retrieve' task. This searches the archive catalogue to find all the 'to be recalled' files, and makes a list of them sorted by location (i.e. by tape reel). Helper asks the operator to mount the tape which contains the most 'to be recalled' files, then copies them back to disk and updates their Catalog entries. It repeats this until all the files have been restored to disk. This means that if you get a "file on tape" response to a recall command, your file should be back on disk within one working day. You can then recall it in the ordinary way. Note that Archiver and Helper make no attempt to send you a copy of the file when it is retrieved from tape.

3.2 EXPIRY OF FILES

Once a week Helper checks the expiry date for all the files in archive. Every user with files which expire within the next two weeks is sent an 'expiry warning'. This take the form of a disk file called ARCHIVE WARNING, which is DISK DUMPed to the user's virtual card reader as a CLASS W file. To read the warning you can

- Execute the archive CHECK comand. If there is a warning file in your reader this DISK LOADs it onto your A disk, then invokes BROWSE to display it. Having read it you QUIT Browse (hit PF3) to return to CMS. If you don't have a warning file in your reader, archive CHECK just returns with a zero return code.

- Use the RDR exec to read the file (possibly giving it a new name), then BROWSE it for yourself.

The 'expiry warning' contains a line for each of your expiring files. Any which are past their expiry dates are marked with stars (*****); they have been expunged. Those which expire in one week or less are marked with plus signs (++++); any unmarked files will expire within two weeks.

The above system works well if you are using the computer regularly, but if not you won't see the warning file.

To cope with this situation Archiver also prints a copy of the warning, and this is posted to you. Overall you get two weeks warning, less the time required to get the warning to you through the mail.

NOTES.

Observe the two-component file names, such as TESTPROG FORTRAN. CMS had no hierarchical directory structure; the only extension was to add the name of another disc, as in SAMPLE DATA B. Even this was done oddly, for the third component of the file identifier was called the file mode, and included certain file attributes.

In this archive system, a file might not be copied onto tape for some time. From the safety point of view, that's a flaw.

A virtual machine in the system described is closely analogous to a process in most other systems, though the implementation is rather different. The intention behind the system design is to present an image of a complete computer as the system metaphor – unusual in that the metaphor is closer to the system hardware than the implementation, whereas most more recent metaphors are designed to get further away from the technicalities.

The "virtual card files" are a consequence of the system metaphor. If your image of the computer is that of a completely isolated machine, then to pass information to another machine you have to punch out a deck of cards. (This was a few years ago.) The "virtual card file" was produced by your "virtual card punch", and would normally be directed to the "virtual card reader" of another virtual machine. It worked, but slowly. The IUCV mentioned was a confession of defeat, and didn't fit into the metaphor at all well. (We introduced virtual card files and IUCV while discussing spooling in the chapter *DISCS – THE SOFTWARE VIEW*.)

REFERENCES.

IMP30 : A. Anderson : *Vax Archiver (Univault) user's guide*, Auckland University Computer Centre C.C. Note #54, 1987.

IMP31 : J.N. Brownlee : *IBM4341 Archiver user's guide*, Auckland University Computer Centre C.C. Note #55, 1987.

QUESTIONS.

What sort of information must the archive system keep about files in archive ? Where should it be kept ?

What must be done if an archive tape is found to be unreadable ? (HINT : define an "invariant" for the archive system, and work out how to restore it.)

Consider ways to implement the required security provisions for the archive's reserved area in the file store.
