Thompson shell

The **Thompson shell** was the first Unix shell, introduced in the first version of Unix in 1971, and was written by Ken Thompson.[1] It was a simple command interpreter, not designed for scripting, but nonetheless introduced several innovative features to the command-line interface and led to the development of the later Unix shells.

Thompson shell

Original author(s)	Ken Thompson
Developer(s)	AT&T Bell Laboratories
Initial release	November 3, 1971
Operating system	Unix and Unix-like
Type	Unix shell

History

The name "shell" for a command-line interpreter and the concept of making the shell a user program outside of the operating system kernel were introduced in Unix's precursor Multics.

An early feature of the Thompson shell was a compact syntax for input/output redirection. In Multics, redirecting the input or output of a command required separate commands to start and stop redirection; in Unix, one could simply add an argument to the command line consisting of the < symbol followed by a filename for input or the > symbol for output, and the shell would redirect I/O for the duration of the command. This syntax was already present by the release of the first version of Unix in 1971.

A later addition was the concept of pipes. At the suggestion of Douglas McIlroy, the redirection syntax was expanded so that the output of one command could be passed to the input of another command. The original pipe syntax, as described in the Version 3 manual, was:

```
command1 >command2>
```

This syntax proved too ambiguous and was easily confused with redirection to and from files—the system cannot tell if "command2" is the command "command2" or the file "command2". By Version 4, the syntax had changed to use both the | and ^ symbols to denote pipes:

```
command1 | command2
```

This produces exactly the same result as:

```
command1 ^ command2
```

The > symbol changed into:

```
command1 > file1
```

This would put the output of command1 into file1.

The Thompson shell syntax for redirection with < and >, and piping with |, has proven durable and has been adopted by most other Unix shells and command shells of several other operating systems, most notably on DOS, OS/2 and Microsoft Windows.

Design

The shell's design was intentionally minimalistic; even the *if* and *goto* statements, essential for control of program flow, were implemented as separate commands.[1]

The shell has no facilities for comments besides a builtin command `:`. Programmers simply write text after this command, which ignores all parameters and simply succeeds. Other builtins include `chdir`, `exit`, `login`, `newgrp`, `shift`, and `wait`.[2]

The *if* command combines the uses of modern-day Bourne shell *test* and *if*. The command first looks for an expression (which can be similar to modern-day *test* or involve an external command) then treats the rest of the command-line as the command to execute if the condition turns out true. There is no *else* branch.[3]

goto is implemented in an interesting way, as it is separate from the shell. When asked to jump to "LABEL", it seeks the current command file for a line that says `: LABEL` (recall that `:` is simply ignored), then exits. When the shell tries to read a next line, the repositioned file descriptor will direct it to the labelled location.[4]

There is no redirection of additional file descriptors other than standard input and output (0 and 1) in Thompson shell. Redirection of stderr (file descriptor 2) also requires an external program wrapper, `fd2`.[5]

The shell supports globbing,[2] but actually implements it by deferring it to a `glob` command that replaces the arguments and calls the requested command.[6]

Thompson shell has positional parameters, but no named variables nor access to environmental variables. It understands the creation of background commands with `&`, similar to Bourne shell. It offers quoting and backslash escapes, though the single quotes work differently from Bourne shell.[2]

Decline and replacements

As a result of the simplistic design, by the 1975 release of Version 6 Unix, it was becoming clear that the Thompson shell was inadequate for most serious programming tasks.

At this time, the developers of the Programmer's Workbench UNIX distribution, most notably John Mashey, began modifying the Thompson shell to make it more suitable for programming.[1] The result, known as the PWB shell or the Mashey shell, included more advanced flow-control mechanisms and introduced shell variables, but remained limited by the necessity to remain compatible with the Thompson shell.

Finally, the Thompson shell was replaced as the main Unix shell by the Bourne shell in Version 7 Unix and the C shell in 2BSD, both released in 1979. Since virtually all modern Unix and Unix-like systems are descended from V7 and 2BSD, the Thompson shell is generally no longer used. It is, however, available as open-source as part of several Ancient Unix source distributions, and has been ported to modern Unices as a historical exhibit.

See also

- Comparison of command shells

References

1. J. R. Mashey (1976-10-13). *Using a Command Language as a High-Level Programming Language* (https://grosskurth.ca/bib/1976/mashey-command.pdf) (PDF). 2nd International Conference on Software Engineering. pp. 169–176.
2. "tsh(1) (html) - TSH(1) - Manuals - Etsh Project (V6Shell)" (https://etsh.nl/man/_tsh.1.html). *etsh.nl*.
3. "if(1) (html) - IF(1) - Manuals - Etsh Project (V6Shell)" (https://etsh.nl/man/_if.1.html). *etsh.nl*.
4. "goto(1) (html) - GOTO(1) - Manuals - Etsh Project (V6Shell)" (https://etsh.nl/man/_goto.1.html). *etsh.nl*.
5. "fd2(1) (html) - FD2(1) - Manuals - Etsh Project (V6Shell)" (https://etsh.nl/man/_fd2.1.html). *etsh.nl*.
6. "glob(1) (html) - GLOB(1) - Manuals - Etsh Project (V6Shell)" (https://etsh.nl/man/_glob.1.html). *etsh.nl*.

External links

- Manual page for the Thompson shell in Unix 1st Edition (http://man.cat-v.org/unix-1st/1/sh).
- The Evolution of the Unix Time-Sharing System (https://www.bell-labs.com/usr/dmr/www/hist.html) - describes the early development of the shell
- Origins of the Bourne shell (http://www.in-ulm.de/~mascheck/bourne/index.html#origins) - manual pages for the 3rd, 4th, and 6th edition Thompson shells, and other resources on the early shells
- Etsh (V6Sh) Project (https://etsh.nl/) - the Thompson shell and its associated utilities ported to modern Unix systems (plus an backwards-compatible "enhanced version" with some modern features)
