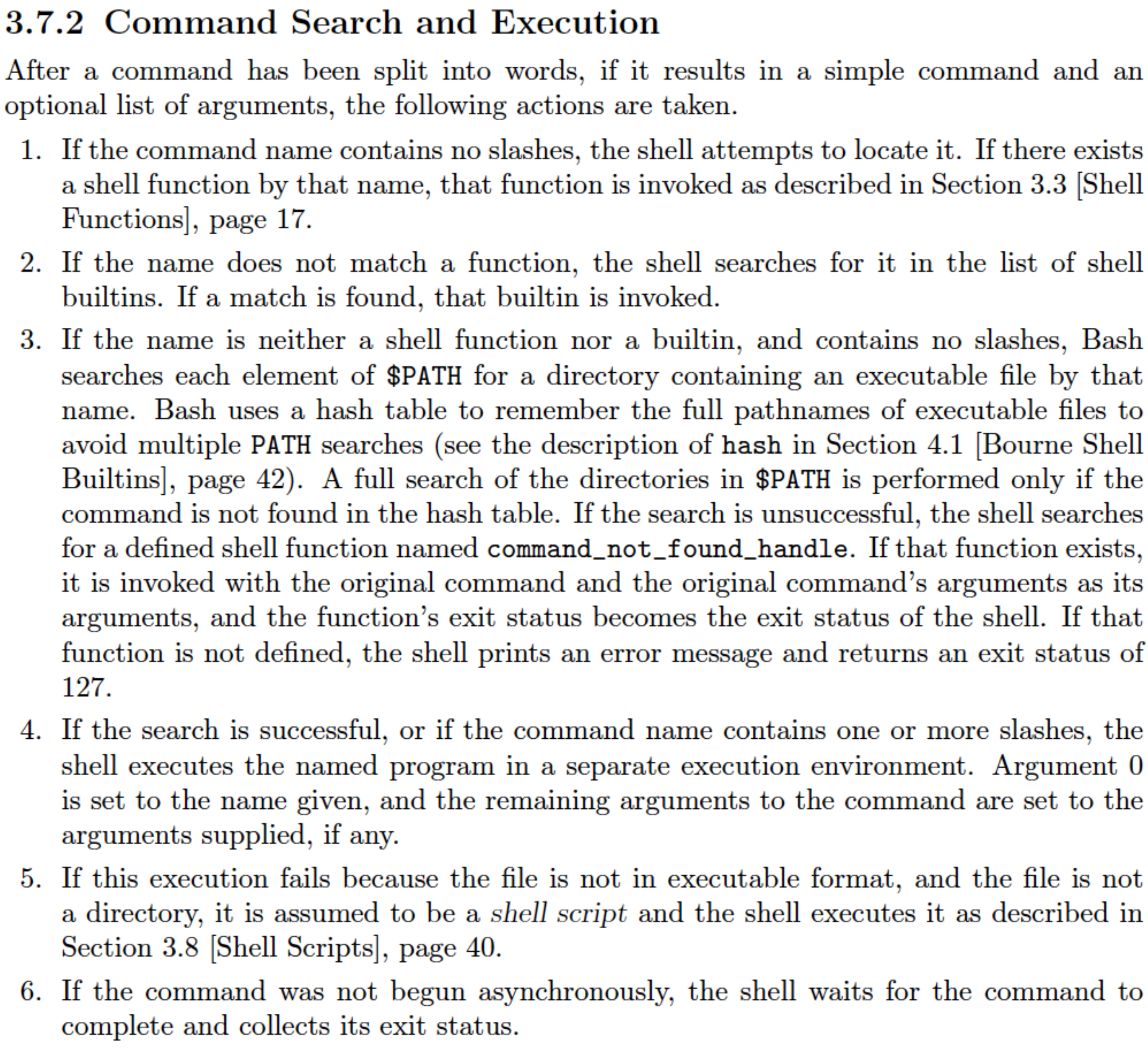
SISTEMAS OPERATIVOS

Recopilacion de conceptos clave

Búsqueda de órdenes y su ejecución – papel de la lista de búsqueda $PATH

En el documento https://www.gnu.org/s/bash/manual/bash.pdf podemos ver una descripción de los pasos seguidos por el intérprete de órdenes cuando ha de ejecutarse una línea de órdenes (Página 37):



3.7.2 Command Search and Execution

After a command has been split into words, if it results in a simple command and an optional list of arguments, the following actions are taken.

1. If the command name contains no slashes, the shell attempts to locate it. If there exists a shell function by that name, that function is invoked as described in Section 3.3 [Shell Functions], page 17.
2. If the name does not match a function, the shell searches for it in the list of shell builtins. If a match is found, that builtin is invoked.
3. If the name is neither a shell function nor a builtin, and contains no slashes, Bash searches each element of `$PATH` for a directory containing an executable file by that name. Bash uses a hash table to remember the full pathnames of executable files to avoid multiple `PATH` searches (see the description of `hash` in Section 4.1 [Bourne Shell Builtins], page 42). A full search of the directories in `$PATH` is performed only if the command is not found in the hash table. If the search is unsuccessful, the shell searches for a defined shell function named `command_not_found_handle`. If that function exists, it is invoked with the original command and the original command's arguments as its arguments, and the function's exit status becomes the exit status of the shell. If that function is not defined, the shell prints an error message and returns an exit status of 127.
4. If the search is successful, or if the command name contains one or more slashes, the shell executes the named program in a separate execution environment. Argument 0 is set to the name given, and the remaining arguments to the command are set to the arguments supplied, if any.
5. If this execution fails because the file is not in executable format, and the file is not a directory, it is assumed to be a *shell script* and the shell executes it as described in Section 3.8 [Shell Scripts], page 40.
6. If the command was not begun asynchronously, the shell waits for the command to complete and collects its exit status.