



Adapted from the document [Project Shell](http://williamstallings.com/OS/OS7e.html); available on <http://williamstallings.com/OS/OS7e.html>

## Your Own Shell

Write a small program that loops reading a line from standard input and checks the first word of the input line. If the first word is one of the following internal commands (or aliases) perform the designated task. Otherwise use the standard ANSI C `system` (Check the [CStdLib.html#system](#) file in the reference files) function to execute the line through the default system shell.

### Internal Commands/Aliases:

#### `clr`

clear the screen using the system function `clear`: `system("clear")`.

#### `dir <directory>`

list the current directory contents (`ls -al <directory>`) - you will need to provide some command line parsing capability to extract the target directory for listing. Once you have built the replacement command line, use the `system` function to execute it.

#### `environ`

list all the environment strings - the environment strings can be accessed from within a program by specifying the POSIX compliant environment list:

```
extern char **environ;
```

as a global variable. `environ` is an array of pointers to the environment strings terminated with a NULL pointer. (see [environ.c "environ.c.html"](#) for examples of use)

#### `quit`

quit from the program with a zero return value. Use the standard `exit` function.

You might want to design your program to make this list of '*aliases*' extendable by storing the alias strings and the alias functions in arrays...

## External Commands:

For all other command line inputs, relay the command line to the parent shell for execution using the `system` function.

When parsing the command line you may have to explicitly or implicitly `malloc` – “CStdLib.html#malloc” (`strdup` – “CStdLib.html#strdup”) storage for a copy of the command line. Ensure that you free any `malloced` memory after it is no longer needed. You may find `strtok` – “CStdLib.html#strtok” useful for parsing.

The C Standard Library has a number of other string related functions that you may find useful (`string.h` – “CStdLib.html#string.h”) contains links to descriptions of the other main "string" functions).

The source of the basis for a simple shell using `strtok` and `system` is contained in `strtokeg.c`. (`strtokeg.c.html`)

Note the number, type and style of comments in `strtokeg.c` - this is the level of commentary expected of the code you hand in for your projects.

Code should be in 'straight' C using the compiler of your choice (`cc` or `gcc`).

Always use `nice` to execute your test programs at lower priority to ensure they do not inconvenience other users if they go 'haywire'. e.g.:

```
>nice a.out
```