Hands On Lab: Process management

Juanjo Costa <jcosta@ac.upc.edu>

Sun, 14 May 2023

Contents

Goals

- Understand and use the file access interface
- Understand and use the absolute and the relative path
- Understand that the file contents are bytes
- Be able to save information in a file and retrieve it
- Be able to search and add information to a file

Use the file access interface

In class we saw the file access interface using the [SDL], let's practice it.

- Make a program (call it *write_string.c*) that asks for a string and save it in a "MYSTRING.TXT" file.

After running this program . . .

- Indicate the absolute path of the "MYSTRING.TXT" file on your machine.
- Indicate the path of the "MYSTRING.TXT" file relative to the directory where your program is located.
- Open a text editor (notepad, for example) and open the "MYSTRING.TXT" file. What do you see there?

- Make a program *write_ints.c* that asks the user for 10 numbers and save them in a “MYINTS.TXT” file. In particular we want to save numbers in binary format as it takes up less space than if we store them in string format. That is, do something like this:

```
int n;
SDL_RWwrite(f, &n, sizeof(n), 1);
```

After running this program . . .

- Open a text editor (notepad, for example) and open the file “MYINTS.TXT”. What do you see there? Oh, surprise! Notepad reads the file and interprets the contents as an ASCII character string, but there are no such “characters” in this file.
- Make a program * read_ints.c * that reads the file “MYINTS.TXT” and displays the saved integers.

Hall of Fame

One of the problems with files is that they only allow you to add new information at the end, but if you try to “add” new information in the middle, the result is that the old content is overwritten. But this is a fairly common use, for example, in our video game context, a typical operation is to save the list of users with the highest scores:

```
Name            Score
-------------------------
MegaZork        666
DepicableMe     10
```

So if a new SuperGirl user arrives with 42 points the list should look like this:

```
Name            Score
-------------------------
MegaZork        666
SuperGirl       42
DepicableMe     10
```

- Implement a *hall_of_fame.c* program that asks for a username (char *) and a score (int), and save it in an orderly manner (by score) to a* FAME.DAT* file. Note that there is not a single type of data here but two different mixed types. Integer types have fixed size, but the length of the character string is unknown and therefore you will need to think of some way to retrieve the different strings (a separator, encode the length before, . . .). Typically, character strings ends with a ‘\0’ character, but storing this content means that in order to retrieve it, we need to read different times looking for that character. A more efficient way could be to first store the length of the string and then the string content. This way, we can retrieve the string with a couple of reads.

References

..[SDL]: http://wiki.libsdl.org/CategoryIO “File I/O Abstraction”