

Literate programming

Yaxley Peaks

August 20, 2022

Literate Programming

Here's Wikipedia on literate programming:

Literate programming is a programming paradigm introduced by Donald Knuth in which a computer program is given an explanation of its logic in a natural language, such as English, interspersed (embedded) with snippets of macros and traditional source code, from which compilable source code can be generated. The approach is used in scientific computing and in data science routinely for reproducible research and open access purposes. Literate programming tools are used by millions of programmers today.

This package however, does not enable complete literate programming. Instead it chooses to go the [haskell route](#) and enable partial literate programming.

1 Getting Started

1.1 Installation

```
git clone https://github.com/yaxley-peaks/unlit
cd unlit
cargo install --path .
```

1.2 Usage

```
unlit -i <IN-FILE> [-o <OUT-FILE>] [-c <COMMENT-SYMBOLS>]
```

Where:

- **IN-FILE**: The input file. (Usually a `.tex`, such as this.)
- **OUT-FILE**: The output file. Writes to `stdout` if none provided.
- **COMMENT-SYMBOLS**: Whatever symbols introduce a single line comment in your language. Defaults to `//` if none are provided.

The input file will usually be a \LaTeX file and truly, any \LaTeX file is allowed. Just enclose all source code in a `code` environment.

1.3 Notes

<IN-FILE> can be any file. As long as all the source code is between `\begin{code}` and `\end{code}`.

2 Example

We will be writing a Bubble sort program in C to demonstrate our literate abilities.

2.1 Setting up

```
#include<stdio.h>
void swap(int *xp, int *yp) {
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}
```

After the obligatory includes, the `swap` function is just a utility function to stream line the value swapping process. Given two pointers to `ints`, it swaps the underlying numbers.

```
void bubbleSort(int arr[], int n) {
```

The function takes an array of `ints` and the length of the array and sorts it in place.

```
    int i, j;
    for (i = 0; i < n - 1; i++)
        for (j = 0; j < n - i - 1; j++)
```

We loop over the array in this fashion and,

```
        if (arr[j] > arr[j + 1])
            swap(&arr[j], &arr[j + 1]);
```

Swap the elements if the next one is bigger than the current one.

```
    }
```

This is the entire algorithm. All code following this is just driver functions.

```
int main(){
    int a[5] = {3,2,4,1,5};
    bubbleSort(a,5);
    for(int i = 0; i < 5; i++){
        printf("%d\n",a[i]);
    }
}
```

And we are done. This file will generate a valid C program. Try it! Run (assuming you have installed `unlit` and are in the root directory of this app.):

```
unlit -i examples/literate.tex -o bubblesort.c
gcc -o bsort bubblesort.c
./bsort
```

Appendix 1

This doesn't actually have all the features of literate haskell. For example: There is no way to ignore code in the literate form. This may or may not change soon. Do not rely on this package. I wrote it when I was bored.

However, pull requests are welcome.