Build Your Own Lisp

Eric Bailey

May 10, 2018 [1]

[1] Last updated May 11, 2018

Write an abstract

Contents

Prompt

1a ⟨*Print version and exit information.* 1a⟩≡
```
puts("Lispy v0.0.1");
puts("Press ctrl-c to exit\n");
```
This code is used in chunk 2a.

1b ⟨*prompt.c* 1b⟩≡
  ⟨*Include the boolean type and values.* 2b⟩
  ⟨*Include the standard I/O functions.* 2c⟩
  ⟨*Include the standard library definitions.* 2d⟩

  ⟨*Include the line editing functions from libedit.* 2e⟩

This definition is continued in chunks 1e and 2a.
Root chunk (not used in this document).

1e ⟨*prompt.c* 1b⟩+≡
```
bool eval(char *input)
{
    if (⟨input is nonempty 1c⟩) {
        ⟨add input to the history table 1d⟩
        printf("< %s\n", input);
    }
    // N.B. This is a no-op when !input.
    free(input);

    return (bool) input;
}
```

Defines:
  `eval`, used in chunk 2a.
Uses `bool` 2b, `free` 2d, and `printf` 2c.

Here, `input` is functionally equivalent to `input` ≠ `NULL`, and `*input` is functionally equivalent to `input[0]` ≠ `'\0'`, i.e. `input` is non-null and nonempty, respectively.

1c ⟨`input` *is nonempty* 1c⟩≡
```
input && *input
```
This code is used in chunk 1e.

1d ⟨*add* `input` *to the history table* 1d⟩≡
```
add_history(input);
```
Uses `add_history` 2e.
This code is used in chunk 1e.

2a ⟨*prompt.c* 1b⟩+≡
```
int main(int argc, char *argv[])
{
    ⟨Print version and exit information. 1a⟩

    while (eval(readline("> ")))
        continue;

    return 0;
}
```
Uses `eval` 1e and `readline` 2e.

Headers

2b ⟨*Include the boolean type and values.* 2b⟩≡
```
#include <stdbool.h>
```
Defines:
`bool`, used in chunk 1e.
This code is used in chunk 1b.

2c ⟨*Include the standard I/O functions.* 2c⟩≡
```
#include <stdio.h>
```
Defines:
`printf`, used in chunk 1e.
This code is used in chunk 1b.

2d ⟨*Include the standard library definitions.* 2d⟩≡
```
#include <stdlib.h>
```
Defines:
`free`, used in chunk 1e.
This code is used in chunk 1b.

2e ⟨*Include the line editing functions from libedit.* 2e⟩≡
```
#include <editline/readline.h>
```
Defines:
`add_history`, used in chunk 1d.
`readline`, used in chunk 2a.
This code is used in chunk 1b.

Chunks

Index

Todo list