Weak language draft

epoll-reactor

December 2021

Contents

1 Scope

This document describes requirements for implementation of weak programming language.

2 Grammar summary

⟨*program*⟩ ::= ⟨*function-declaration*⟩*

⟨*function-declaration*⟩ ::= ⟨*type*⟩ ⟨*id*⟩ ***(*** ⟨*parameter-list-opt*⟩ ***)*** **{** ⟨*stmt*⟩* **}**

⟨*type*⟩ ::= ***int***
| ***char***
| ***string***
| ***boolean***
| ***void***

⟨*alpha*⟩ ::= *(* ***a*** | ***b*** | ... | ***z*** | _ *)*

⟨*digit*⟩ ::= ***0*** | ***1*** | ... ***9***

⟨*boolean-literal*⟩ ::= ***true***
| ***false***

⟨*integral-literal*⟩ ::= ⟨*digit*⟩*

⟨*floating-literal*⟩ ::= ⟨*digit*⟩* **.** ⟨*digit*⟩*

⟨*constant*⟩ ::= ⟨*integral-literal*⟩
| ⟨*floating-literal*⟩
| ⟨*string-literal*⟩
| ⟨*boolean-literal*⟩

⟨*id*⟩ ::= ⟨*alpha*⟩ **(** ⟨*alpha*⟩ | ⟨*digit*⟩ **)***

⟨*parameter*⟩ ::= ⟨*type*⟩ ⟨*id*⟩

⟨*parameter-list*⟩ ::= ⟨*parameter*⟩ **,** ⟨*parameter-list*⟩
| ⟨*parameter*⟩

⟨*parameter-list-opt*⟩ ::= ⟨*parameter-list*⟩ | ϵ

⟨*stmt*⟩ ::= ⟨*selection-stmt*⟩
| ⟨*iteration-stmt*⟩
| ⟨*jump-stmt*⟩
| ⟨*expr*⟩

⟨*selection-stmt*⟩ ::= `if (` ⟨*expr*⟩ `) {` ⟨*stmt*⟩`* }`
| `if (` ⟨*expr*⟩ `) {` ⟨*stmt*⟩`* } else {` ⟨*stmt*⟩`* }`

⟨*iteration-stmt*⟩ ::= `for (` ⟨*expr*⟩`? ;` ⟨*expr*⟩`? ;` ⟨*expr*⟩`? ) {` ⟨*stmt*⟩`* }`
| `while (` ⟨*expr*⟩ `) {` ⟨*stmt*⟩`* }`
| `do {` ⟨*stmt*⟩`* } while (` ⟨*expr*⟩`* )`

⟨*jump-stmt*⟩ ::= `return` ⟨*expr*⟩`? ;`

⟨*assignment-op*⟩ ::= `=`
| `*=`
| `/=`
| `%=`
| `+=`
| `-=`
| `<<=`
| `>>=`
| `&=`
| `|=`
| `^=`

⟨*expr*⟩ ::= ⟨*assignment-expr*⟩

⟨*assignment-expr*⟩ ::= ⟨*logical-or-expr*⟩
| ⟨*unary-expr*⟩ ⟨*assignment-op*⟩ ⟨*assignment-expr*⟩

⟨*logical-or-expr*⟩ ::= ⟨*logical-and-expr*⟩
| ⟨*logical-or-expr*⟩ `||` ⟨*logical-and-expr*⟩

⟨*logical-and-expr*⟩ ::= ⟨*inclusive-or-expr*⟩
| ⟨*logical-and-expr*⟩ `&&` ⟨*inclusive-or-expr*⟩

⟨*inclusive-or-expr*⟩ ::= ⟨*exclusive-or-expr*⟩
| ⟨*inclusive-or-expr*⟩ `|` ⟨*exclusive-or-expr*⟩

⟨*exclusive-or-expr*⟩ ::= ⟨*and-expr*⟩
| ⟨*exclusive-or-expr*⟩ `^` ⟨*and-expr*⟩

⟨*and-expr*⟩ ::= ⟨*equality-expr*⟩
| ⟨*and-expr*⟩ `&` ⟨*equality-expr*⟩

⟨*equality-expr*⟩ ::= ⟨*relational-expr*⟩
| ⟨*equality-expr*⟩ `==` ⟨*relational-expr*⟩
| ⟨*equality-expr*⟩ `!=` ⟨*relational-expr*⟩

⟨*relational-expr*⟩ ::= ⟨*shift-expr*⟩
| ⟨*relational-expr*⟩ `>` ⟨*shift-expr*⟩
| ⟨*relational-expr*⟩ `<` ⟨*shift-expr*⟩
| ⟨*relational-expr*⟩ `>=` ⟨*shift-expr*⟩
| ⟨*relational-expr*⟩ `<=` ⟨*shift-expr*⟩

⟨*shift-expr*⟩ ::= ⟨*additive-expr*⟩
| ⟨*shift-expr*⟩ `<<` ⟨*additive-expr*⟩
| ⟨*shift-expr*⟩ `>>` ⟨*additive-expr*⟩

⟨*additive-expr*⟩ ::= ⟨*multiplicative-expr*⟩
| ⟨*additive-expr*⟩ `+` ⟨*multiplicative-expr*⟩
| ⟨*additive-expr*⟩ `-` ⟨*multiplicative-expr*⟩

⟨*multiplicative-expr*⟩ ::= ⟨*unary-expr*⟩
| ⟨*multiplicative-expr*⟩ `*` ⟨*unary-expr*⟩
| ⟨*multiplicative-expr*⟩ `/` ⟨*unary-expr*⟩
| ⟨*multiplicative-expr*⟩ `%` ⟨*unary-expr*⟩

⟨*unary-expr*⟩ ::= ⟨*postfix-expr*⟩
| `++` ⟨*unary-expr*⟩
| `--` ⟨*unary-expr*⟩

⟨*postfix-expr*⟩ ::= ⟨*primary-expr*⟩
| ⟨*postfix-expr*⟩ `[` ⟨*expr*⟩ `]`
| ⟨*postfix-expr*⟩ `++`
| ⟨*postfix-expr*⟩ `--`

⟨*primary-expr*⟩ ::= ⟨*constant*⟩
| ⟨*id*⟩
| `(` ⟨*expr*⟩ `)`

3 Environment

3.1 Translation environment

The whole program must be placed in one file to simplify translation and linking (lack of it as such).