C48 Master Branch 10/5/17 5pm

Purpose

This document primarily aims to organize our functions in a small and easy to use database. The files in this document are located in the /src folder where most of our code is. Updating this document after a push is highly recommended, but not necessary. When updated, make sure to update the header.

Key

Struct
Enum
Function
Global Var

U - Uncommented/Needs better Documentation
H - Hard to Understand
F - Needs Formatting
D - Duplicate Function

---

compiler.c

2 Functions

	void compile()
	void assembler()

---

eval-apply.c

5 Functions

U	int self_evaluatingp(object *exp)
U	int primitivep(object *exp)
U	object *apply_primitive_procedure(object *procedure, object *arguments)
	char *apply(char operator, int arguments[])
	char *eval(eval_arguments exp_env)

---

identifier.c

1 Struct, 2 Global Vars, 2 Functions

	struct identifier

C48 Master Branch 10/5/17 5pm

	static char *identifier_string
	static double number_value
	struct identifier *read_identifier(char *program, int index)
	struct identifier *read_number(char *program, int index)

---

lexer.c
2 Functions

	token_list* list_lexer(char *program)
	token_list* list_lexer_tmp(char *program)

---

pair.c
5 Structs, 15 Functions

	struct token
	struct object
	typedef struct token_list
	token_list* create_token(struct token token, token_list *next)
	token_list* prepend(struct token token, token_list *head)
	int count_tokenlist(token_list *head)
	token_list* reverse_tokenlist(token_list *head)
	typedef struct pair_cell
	typedef struct pair_token
	void print(struct pair_token *list)
	pair_cell* create1(void *car, void *cdr)
	char* car(struct pair_token *list)
	pair_token* cdr(struct pair_token* list)
	pair_token* cons(void *car, pair_token *cdr)
	pair_cell* cons1(struct object val, struct pair_cell *cdr)

C48 Master Branch 10/5/17 5pm

	int count_nodes1(pair_cell *head)
	int count_nodes(pair_token *head)
	pair_cell* reverse_code_tree(pair_cell *head)
	pair_cell* remove_front(pair_cell *head)
	pair_cell* read_from_tokens(struct pair_cell *token_list)

---

parser.c
2 Structs, 9 Functions

	typedef struct object
	object* cons(object *car, object *cdr)
	object* car(object *cell)
	object* cdr(object *cell)
	object* create_number(int number)
	object* create_variable(char* variable)
	object* create_primativeop(char* variable)
	typedef struct type_list
	char* get_car(void *car)
	void print_token_list2(token_list *token_list)
	object* parse(token_list *token_list, object *code_tree)

---

print.c
1 Function

	char* print(object *result)

---

read.c
2 Structs, 2 Global Vars, 1 Function

	static char *identifier_string

C48 Master Branch 10/5/17 5pm

	static double number_value
	typedef struct eval_arguments_token
	typedef struct eval_arguments_cell
	struct eval_arguements1 parser(struct pair_cell *token_list)

---

read2.c
5 Global Vars, 1 Struct, 4 Functions

	int left
	int right
	int invalid
	int value
	char charSet[]
	struct Token
	int isnumber(char s)
	int isoperator(char s)
	int isbrackets(char s)
	char* read_token(char *program)

---

read_o.c
1 Enum, 2 Global Vars, 2 Structs, 7 Functions

	enum Token
	static char *identifier_string
	static double number_value
	typedef struct pair
	typedef struct eval_arguements
	pair* create1(void *car, void *cdr)
	pair* cons(void *car, pair *cdr)

C48 Master Branch 10/5/17 5pm

	int isnumber(char *s)
	struct eval_arguements read(char *program)
	char* read_token(char *program)
	int read_list(pair *list_so_far)
	char* micro_read(char *program)

---

repl.c
Main

	int main(char *argc, char **argv[])

---

token.c
2 Structs, 10 Functions

	struct token_object
	typedef struct token_list
	char* token_type(char *token)
	token_list* reverse_tokenlist(token_list *head)
	token_list* prepend_token(struct token_object val, struct token_list *cdr)
	int count_token_list(token_list *cursor)
	char* first(struct token_list *list)
	char* find_value(struct token_list *list)
	char* find_type(struct token_list *list)
	token_list* rest(struct token_list *list)
	char* print_token_list(struct token_list *list, char *result)
	char* print_token_list_debug(struct token_list *list, char *result)

---

utils.c
6 Functions

C48 Master Branch 10/5/17 5pm

	char* chopN(char *charBuffer, int n)
	char* scat(char *s, char *t)
	int iswhitespace(char c)
	char* append(char *s, char c)
	int count_chars(char *string, char ch)
	int isnumber(char *s)

---

vm.c
3 Global Vars, 6 Functions

	int MAXSIZE
	int stack[8];
	int top
	int isEmpty()
	int isFull()
	int peek()
	int pop()
	int push(int data)
	void machine(int code[])

Other files in src

	Makefile
	vm.h
	ztwild(Hello)