My Project

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 node Struct Reference

`#include <blank.h>`

Collaboration diagram for node:

Public Attributes

- struct node * child_head
- char * name
- struct node * next
- struct node * parent
- int parentheses
- struct node * prev

3.1.1 Member Data Documentation

3.1.1.1 child_head

`struct node* node::child_head`

3.1.1.2 name

`char* node::name`

3.1.1.3 next

```
struct node* node::next
```

3.1.1.4 parent

```
struct node* node::parent
```

3.1.1.5 parentheses

```
int node::parentheses
```

3.1.1.6 prev

```
struct node* node::prev
```

The documentation for this struct was generated from the following file:

- blank.h

3.2 operator_precedence Struct Reference

```
#include <blank.h>
```

Public Attributes

- char * operator
- int precedence

3.2.1 Member Data Documentation

3.2.1.1 operator

```
char* operator_precedence::operator
```

3.2.1.2 precedence

```
int operator_precedence::precedence
```

The documentation for this struct was generated from the following file:

- blank.h

3.3 ssu_scoreTable Struct Reference

```
#include <ssu_score.h>
```

Public Attributes

- char qname [FILELEN]
- double score

3.3.1 Member Data Documentation

3.3.1.1 qname

```
char ssu_scoreTable::qname[FILELEN]
```

3.3.1.2 score

```
double ssu_scoreTable::score
```

The documentation for this struct was generated from the following file:

- ssu_score.h

Chapter 4

File Documentation

4.1 blank.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
#include <ctype.h>
#include "blank.h"
```

Include dependency graph for blank.c:

4.2 blank.h File Reference

This graph shows which files directly or indirectly include this file:

Classes

- struct node
- struct operator_precedence

Macros

- #define BUFLEN 1024
- #define DATATYPE_SIZE 35
- #define false 0
- #define MINLEN 64
- #define OPERATOR_CNT 24
- #define TOKEN_CNT 50
- #define true 1

Typedefs

- typedef struct node node
- typedef struct operator_precedence operator_precedence

Functions

- int all_character (char *str)
- int all_star (char *str)
- node * change_sibling (node *parent)
- int check_brackets (char *str)
- void clear_tokens (char tokens[TOKEN_CNT][MINLEN])
- void compare_tree (node *root1, node *root2, int *result)
- node * create_node (char *name, int parentheses)
- int find_typeSpecifier (char tokens[TOKEN_CNT][MINLEN])
- int find_typeSpecifier2 (char tokens[TOKEN_CNT][MINLEN])
- void free_node (node *cur)
- node * get_high_precedence_node (node *cur, node *new)
- node * get_last_child (node *cur)
- node * get_most_high_precedence_node (node *cur, node *new)
- node * get_operator (node *cur)
- int get_precedence (char *op)
- node * get_root (node *cur)
- int get_sibling_cnt (node *cur)
- int get_token_cnt (char tokens[TOKEN_CNT][MINLEN])
- node * insert_node (node *old, node *new)
- int is_character (char c)
- int is_operator (char *op)
- int is_typeStatement (char *str)
- char * ltrim (char *_str)
- int make_tokens (char *str, char tokens[TOKEN_CNT][MINLEN])
- node * make_tree (node *root, char(*tokens)[MINLEN], int *idx, int parentheses)
- void print (node *cur)
- char * remove_extraspace (char *str)
- void remove_space (char *str)
- int reset_tokens (int start, char tokens[TOKEN_CNT][MINLEN])
- char * rtrim (char *_str)

4.2.1 Macro Definition Documentation

4.2.1.1 BUFLEN

```
#define BUFLEN 1024
```

4.2.1.2 DATATYPE_SIZE

```
#define DATATYPE_SIZE 35
```

4.2.1.3 false

```
#define false 0
```

4.2.1.4 MINLEN

```
#define MINLEN 64
```

4.2.1.5 OPERATOR_CNT

```
#define OPERATOR_CNT 24
```

4.2.1.6 TOKEN_CNT

```
#define TOKEN_CNT 50
```

4.2.1.7 true

```
#define true 1
```

4.2.2 Typedef Documentation

4.2.2.1 node

```
typedef struct node node
```

4.2.2.2 operator_precedence

```
typedef struct operator_precedence operator_precedence
```

4.2.3 Function Documentation

4.2.3.1 all_character()

```
int all_character (
            char * str )
```

Here is the call graph for this function:

4.2.3.2 all_star()

```
int all_star (
            char * str )
```

Here is the caller graph for this function:

4.2.3.3 change_sibling()

```
node* change_sibling (
            node * parent )
```

Here is the caller graph for this function:

4.2.3.4 check_brackets()

```
int check_brackets (
            char * str )
```

Here is the caller graph for this function:

4.2.3.5 clear_tokens()

```
void clear_tokens (
            char tokens[TOKEN_CNT][MINLEN] )
```

Here is the caller graph for this function:

4.2.3.6 compare_tree()

```
void compare_tree (
            node * root1,
            node * root2,
            int * result )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.7 create_node()

```
node* create_node (
            char * name,
            int parentheses )
```

Here is the caller graph for this function:

4.2.3.8 find_typeSpecifier()

```
int find_typeSpecifier (
            char tokens[TOKEN_CNT][MINLEN] )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.9 find_typeSpecifier2()

```
int find_typeSpecifier2 (
            char tokens[TOKEN_CNT][MINLEN] )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.10 free_node()

```
void free_node (
            node * cur )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.11 get_high_precedence_node()

```
node* get_high_precedence_node (
            node * cur,
            node * new )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.12 get_last_child()

```
node* get_last_child (
            node * cur )
```

Here is the caller graph for this function:

4.2.3.13 get_most_high_precedence_node()

```
node* get_most_high_precedence_node (
            node * cur,
            node * new )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.14 get_operator()

```
node* get_operator (
            node * cur )
```

Here is the caller graph for this function:

4.2.3.15 get_precedence()

```
int get_precedence (
            char * op )
```

Here is the caller graph for this function:

4.2.3.16 get_root()

```
node* get_root (
            node * cur )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.17 get_sibling_cnt()

```
int get_sibling_cnt (
            node * cur )
```

Here is the caller graph for this function:

4.2.3.18 get_token_cnt()

```
int get_token_cnt (
            char tokens[TOKEN_CNT][MINLEN] )
```

4.2.3.19 insert_node()

```
node* insert_node (
            node * old,
            node * new )
```

Here is the caller graph for this function:

4.2.3.20 is_character()

```
int is_character (
            char c )
```

Here is the caller graph for this function:

4.2.3.21 is_operator()

```
int is_operator (
            char * op )
```

Here is the caller graph for this function:

4.2.3.22 is_typeStatement()

```
int is_typeStatement (
            char * str )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.23 ltrim()

```
char* ltrim (
            char * _str )
```

Here is the caller graph for this function:

4.2.3.24 make_tokens()

```
int make_tokens (
            char * str,
            char tokens[TOKEN_CNT][MINLEN] )
```

Here is the caller graph for this function:

4.2.3.25 make_tree()

```
node* make_tree (
            node * root,
            char(*) tokens[MINLEN],
            int * idx,
            int parentheses )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.26 print()

```
void print (
            node * cur )
```

4.2.3.27 remove_extraspace()

```
char* remove_extraspace (
            char * str )
```

Here is the caller graph for this function:

4.2.3.28 remove_space()

```
void remove_space (
            char * str )
```

Here is the caller graph for this function:

4.2.3.29 reset_tokens()

```
int reset_tokens (
            int start,
            char tokens[TOKEN_CNT][MINLEN] )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.3.30 rtrim()

```
char* rtrim (
            char * _str )
```

Here is the caller graph for this function:

4.3 main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/time.h>
#include "ssu_score.h"
```

Include dependency graph for main.c:

Macros

- #define SECOND_TO_MICRO 1000000

Functions

- int main (int argc, char ∗argv[ ])
- void ssu_runtime (struct timeval ∗begin_t, struct timeval ∗end_t)

4.3.1 Macro Definition Documentation

4.3.1.1 SECOND_TO_MICRO

```
#define SECOND_TO_MICRO 1000000
```

4.3.2 Function Documentation

4.3.2.1 main()

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

Here is the call graph for this function:

4.3.2.2 ssu_runtime()

```
void ssu_runtime (
            struct timeval * begin_t,
            struct timeval * end_t )
```

Here is the caller graph for this function:

4.4 ssu_score.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <signal.h>
#include <string.h>
#include <sys/types.h>
#include <dirent.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include "ssu_score.h"
#include "blank.h"
```

Functions

- double check_error_warning (char *filename)
- int check_option (int argc, char *argv[])
- int compare_resultfile (char *file1, char *file2)
- double compile_program (char *id, char *filename)
- void do_iOption (int argc, char *argv[], int optind)
- void do_mOption (char *path)
- int execute_program (char *id, char *filename)
- char * get_answer (int fd, char *result)
- int get_create_type ()
- int get_file_type (char *filename)
- void get_qname_number (char *qname, int *num1, int *num2)
- pid_t inBackground (char *name)
- int is_thread (char *qname)
- void make_scoreTable (char *ansDir)
- void print_usage ()
- void read_scoreTable (char *path)
- void redirection (char *command, int new, int old)
- void rmdirs (const char *path)
- int score_blank (char *id, char *filename)
- double score_program (char *id, char *filename)
- double score_student (int fd, char *id)
- void score_students ()
- void set_idTable (char *stuDir)
- void set_scoreTable (char *ansDir)
- void sort_idTable (int size)
- void sort_scoreTable (int size)
- void ssu_score (int argc, char *argv[])
- void to_lower_case (char *c)
- void write_first_row (int fd)
- void write_scoreTable (char *filename)

Variables

- char ansDir [BUFLEN]
- int eOption = false

  -i
- char errorDir [BUFLEN]
- char id_table [SNUM][10]
- int iOption = false
- int mOption = false
- char printId [ARGNUM][FILELEN]
- struct ssu_scoreTable score_table [QNUM]
- char stuDir [BUFLEN]
- char threadFiles [ARGNUM][FILELEN]

- int tOption = false

4.4.1 Function Documentation

4.4.1.1 check_error_warning()

```
double check_error_warning (
            char * filename )
```

Here is the caller graph for this function:

4.4.1.2 check_option()

```
int check_option (
            int argc,
            char * argv[] )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.3 compare_resultfile()

```
int compare_resultfile (
            char * file1,
            char * file2 )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.4 compile_program()

```
double compile_program (
            char * id,
            char * filename )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.5 do_iOption()

```
void do_iOption (
            int argc,
            char * argv[],
            int optind )
```

Here is the caller graph for this function:

4.4.1.6 do_mOption()

```
void do_mOption (
            char * path )
```

Here is the caller graph for this function:

4.4.1.7 execute_program()

```
int execute_program (
             char * id,
             char * filename )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.8 get_answer()

```
char* get_answer (
             int fd,
             char * result )
```

Here is the caller graph for this function:

4.4.1.9 get_create_type()

```
int get_create_type ( )
```

Here is the caller graph for this function:

4.4.1.10 get_file_type()

```
int get_file_type (
             char * filename )
```

Here is the caller graph for this function:

4.4.1.11 get_qname_number()

```
void get_qname_number (
             char * qname,
             int * num1,
             int * num2 )
```

Here is the caller graph for this function:

4.4.1.12 inBackground()

```
pid_t inBackground (
             char * name )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.13 is_thread()

```
int is_thread (
            char * qname )
```

Here is the caller graph for this function:

4.4.1.14 make_scoreTable()

```
void make_scoreTable (
            char * ansDir )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.15 print_usage()

```
void print_usage ( )
```

Here is the caller graph for this function:

4.4.1.16 read_scoreTable()

```
void read_scoreTable (
            char * path )
```

Here is the caller graph for this function:

4.4.1.17 redirection()

```
void redirection (
            char * command,
            int new,
            int old )
```

Here is the caller graph for this function:

4.4.1.18 rmdirs()

```
void rmdirs (
            const char * path )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.19 score_blank()

```
int score_blank (
            char * id,
            char * filename )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.20 score_program()

```
double score_program (
            char * id,
            char * filename )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.21 score_student()

```
double score_student (
            int fd,
            char * id )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.22 score_students()

```
void score_students ( )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.23 set_idTable()

```
void set_idTable (
            char * stuDir )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.24 set_scoreTable()

```
void set_scoreTable (
            char * ansDir )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.25 sort_idTable()

```
void sort_idTable (
            int size )
```

Here is the caller graph for this function:

4.4.1.26 sort_scoreTable()

```
void sort_scoreTable (
            int size )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.27 ssu_score()

```
void ssu_score (
            int argc,
            char * argv[] )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.28 to_lower_case()

```
void to_lower_case (
            char * c )
```

Here is the caller graph for this function:

4.4.1.29 write_first_row()

```
void write_first_row (
            int fd )
```

Here is the caller graph for this function:

4.4.1.30 write_scoreTable()

```
void write_scoreTable (
            char * filename )
```

Here is the caller graph for this function:

4.4.2 Variable Documentation

4.4.2.1 ansDir

```
char ansDir[BUFLEN]
```

4.4.2.2 eOption

```
int eOption = false
```

-i

4.4.2.3 errorDir

```
char errorDir[BUFLEN]
```

4.4.2.4 id_table

```
char id_table
```

4.4.2.5 iOption

```
int iOption = false
```

4.4.2.6 mOption

```
int mOption = false
```

4.4.2.7 printId

```
char printId[ARGNUM][FILELEN]
```

4.4.2.8 score_table

```
struct ssu_scoreTable score_table
```

4.4.2.9 stuDir

```
char stuDir[BUFLEN]
```

4.4.2.10 threadFiles

```
char threadFiles[ARGNUM][FILELEN]
```

4.4.2.11 tOption

```
int tOption = false
```

4.5 ssu_score.h File Reference

This graph shows which files directly or indirectly include this file:

Classes

- struct ssu_scoreTable

Macros

- #define ARGNUM 5
- #define BUFLEN 1024
- #define CFILE 4
- #define ERROR 0
- #define false 0
- #define FILELEN 64
- #define OVER 5
- #define QNUM 100
- #define SNUM 100
- #define STDERR 2
- #define STDOUT 1
- #define TEXTFILE 3
- #define true 1
- #define WARNING -0.1

Functions

- double check_error_warning (char *filename)
- int check_option (int argc, char *argv[ ])
- int compare_resultfile (char *file1, char *file2)
- double compile_program (char *id, char *filename)
- void do_iOption (int argc, char *argv[ ], int optind)
- int execute_program (char *id, char *filname)
- char * get_answer (int fd, char *result)
- int get_create_type ()
- int get_file_type (char *filename)
- void get_qname_number (char *qname, int *num1, int *num2)
- pid_t inBackground (char *name)
- int is_exist (char(*src)[FILELEN], char *target)
- int is_thread (char *qname)
- void make_scoreTable (char *ansDir)
- void print_usage ()
- void read_scoreTable (char *path)
- void redirection (char *command, int newfd, int oldfd)
- void rmdirs (const char *path)

- int score_blank (char *id, char *filename)
- double score_program (char *id, char *filename)
- double score_student (int fd, char *id)
- void score_students ()
- void set_idTable (char *stuDir)
- void set_scoreTable (char *ansDir)
- void sort_idTable (int size)
- void sort_scoreTable (int size)
- void ssu_score (int argc, char *argv[])
- void to_lower_case (char *c)
- void write_first_row (int fd)
- void write_scoreTable (char *filename)

4.5.1 Macro Definition Documentation

4.5.1.1 ARGNUM

```
#define ARGNUM 5
```

4.5.1.2 BUFLEN

```
#define BUFLEN 1024
```

4.5.1.3 CFILE

```
#define CFILE 4
```

4.5.1.4 ERROR

```
#define ERROR 0
```

4.5.1.5 false

```
#define false 0
```

4.5.1.6 FILELEN

```
#define FILELEN 64
```

4.5.1.7 OVER

```
#define OVER 5
```

4.5.1.8 QNUM

```
#define QNUM 100
```

4.5.1.9 SNUM

```
#define SNUM 100
```

4.5.1.10 STDERR

```
#define STDERR 2
```

4.5.1.11 STDOUT

```
#define STDOUT 1
```

4.5.1.12 TEXTFILE

```
#define TEXTFILE 3
```

4.5.1.13 true

```
#define true 1
```

4.5.1.14 WARNING

```
#define WARNING -0.1
```

4.5.2 Function Documentation

4.5.2.1 check_error_warning()

```
double check_error_warning (
            char * filename )
```

Here is the caller graph for this function:

4.5.2.2 check_option()

```
int check_option (
            int argc,
            char * argv[] )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.3 compare_resultfile()

```
int compare_resultfile (
            char * file1,
            char * file2 )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.4 compile_program()

```
double compile_program (
            char * id,
            char * filename )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.5 do_iOption()

```
void do_iOption (
            int argc,
            char * argv[],
            int optind )
```

Here is the caller graph for this function:

4.5.2.6 execute_program()

```
int execute_program (
             char * id,
             char * filname )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.7 get_answer()

```
char* get_answer (
             int fd,
             char * result )
```

Here is the caller graph for this function:

4.5.2.8 get_create_type()

```
int get_create_type ( )
```

Here is the caller graph for this function:

4.5.2.9 get_file_type()

```
int get_file_type (
             char * filename )
```

Here is the caller graph for this function:

4.5.2.10 get_qname_number()

```
void get_qname_number (
             char * qname,
             int * num1,
             int * num2 )
```

Here is the caller graph for this function:

4.5.2.11 inBackground()

```
pid_t inBackground (
             char * name )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.12 is_exist()

```
int is_exist (
            char(*) src[FILELEN],
            char * target )
```

4.5.2.13 is_thread()

```
int is_thread (
            char * qname )
```

Here is the caller graph for this function:

4.5.2.14 make_scoreTable()

```
void make_scoreTable (
            char * ansDir )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.15 print_usage()

```
void print_usage ( )
```

Here is the caller graph for this function:

4.5.2.16 read_scoreTable()

```
void read_scoreTable (
            char * path )
```

Here is the caller graph for this function:

4.5.2.17 redirection()

```
void redirection (
            char * command,
            int newfd,
            int oldfd )
```

Here is the caller graph for this function:

4.5.2.18 rmdirs()

```
void rmdirs (
            const char * path )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.19 score_blank()

```
int score_blank (
            char * id,
            char * filename )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.20 score_program()

```
double score_program (
            char * id,
            char * filename )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.21 score_student()

```
double score_student (
            int fd,
            char * id )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.22 score_students()

```
void score_students ( )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.23 set_idTable()

```
void set_idTable (
            char * stuDir )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.24 set_scoreTable()

```
void set_scoreTable (
            char * ansDir )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.25 sort_idTable()

```
void sort_idTable (
            int size )
```

Here is the caller graph for this function:

4.5.2.26 sort_scoreTable()

```
void sort_scoreTable (
            int size )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.27 ssu_score()

```
void ssu_score (
            int argc,
            char * argv[] )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.2.28 to_lower_case()

```
void to_lower_case (
            char * c )
```

Here is the caller graph for this function:

4.5.2.29 write_first_row()

```
void write_first_row (
            int fd )
```

Here is the caller graph for this function:

4.5.2.30 write_scoreTable()

```
void write_scoreTable (
            char * filename )
```

Here is the caller graph for this function: