Trahrhe Collapse

Generated by Doxygen 1.9.1

1 Liens utiles 1

1.1 Use case . . . 1

1.2 Todo list (non-exhaustive) . . . 2

2 Dependencies 3

2.1 ## Troubleshooting . . . 3

3 LICENCE 5

4 Usage 7

4.1 Prerequisites . . . 7

4.2 Steps . . . 7

4.3 Example . . . 7

5 Automatic-loop-collapsing 9

5.1 Installation . . . 9

5.2 Feature Definitions . . . 9

5.3 Usage . . . 9

5.4 License . . . 9

6 Data Structure Index 11

6.1 Data Structures . . . 11

7 File Index 13

7.1 File List . . . 13

8 Data Structure Documentation 15

8.1 boundary Struct Reference . . . 15

8.1.1 Detailed Description . . . 15

8.2 boundaryList Struct Reference . . . 15

8.3 iterationDomain Struct Reference . . . 16

8.3.1 Detailed Description . . . 16

8.4 iterationDomainList Struct Reference . . . 16

8.5 TCD_FlowData Struct Reference . . . 16

8.5.1 Detailed Description . . . 16

9 File Documentation 17

9.1 include/codegen.h File Reference . . . 17

9.1.1 Detailed Description . . . 17

9.1.2 Function Documentation . . . 18

9.1.2.1 generateCode() . . . 18

9.1.2.2 generateHeaderFile() . . . 18

9.1.2.3 mergeGeneratedCode() . . . 18

9.2 include/data.h File Reference . . . 18

9.2.1 Detailed Description . . . 19

9.2.2 Function Documentation . . . 19
9.2.2.1 copyBoundary() . . . 19
9.2.2.2 copyIterationDomain() . . . 20
9.2.2.3 getBoundaries() . . . 20
9.2.2.4 getBoundary() . . . 20
9.2.2.5 printBoundaries() . . . 21
9.3 include/flow.h File Reference . . . 21
9.3.1 Detailed Description . . . 22
9.3.2 Function Documentation . . . 22
9.3.2.1 endTcdFlow() . . . 22
9.3.2.2 initTcdFlow() . . . 22
9.4 include/format_helper.h File Reference . . . 22
9.4.1 Detailed Description . . . 23
9.4.2 Function Documentation . . . 23
9.4.2.1 digit_check() . . . 23
9.4.2.2 tabString() . . . 23
9.4.2.3 tabStringReturn() . . . 24
9.4.2.4 take() . . . 24
9.5 include/fs.h File Reference . . . 24
9.5.1 Detailed Description . . . 25
9.5.2 Function Documentation . . . 25
9.5.2.1 fs_open() . . . 25
9.5.2.2 fs_rewind() . . . 25
9.5.2.3 fs_writef() . . . 25
9.5.2.4 fs_writeft() . . . 25
9.6 src/codegen.c File Reference . . . 25
9.7 src/flow.c File Reference . . . 26
9.7.1 Detailed Description . . . 26
9.7.2 Function Documentation . . . 26
9.7.2.1 endTcdFlow() . . . 26
9.7.2.2 initTcdFlow() . . . 26
9.8 src/format_helper.c File Reference . . . 27
9.8.1 Detailed Description . . . 27
9.8.2 Function Documentation . . . 27
9.8.2.1 digit_check() . . . 27
9.8.2.2 tabString() . . . 27
9.8.2.3 tabStringReturn() . . . 28
9.8.2.4 take() . . . 28
9.9 src/fs.c File Reference . . . 28
9.9.1 Detailed Description . . . 29
9.9.2 Function Documentation . . . 29
9.9.2.1 fs_open() . . . 29

9.9.2.2 fs_rewind() 29
9.9.2.3 fs_writef() 29
9.9.2.4 fs_writeft() 29

Index **31**

Chapter 1

Liens utiles

- Benchmarks: polybench https://web.cs.ucla.edu/~pouchet/software/polybench/
- Openscop: https://icps.u-strasbg.fr/people/bastoul/public_html/development/openscop/html
- Clan: https://icps.u-strasbg.fr/~bastoul/development/clan/
- Trahrhe: https://webpages.gitlabpages.inria.fr/trahrhe/documentation
- Cloog: http://www.bastoul.net/cloog/
- (Atiling: https://github.com/Zetsyog/atiling)

1.1 Use case

```
# exemple d'usage
(collapse) [input.c] -o [output.c] # des options supplémentaires sont envisageables
```

Avec `input.c` de la forme

```
/*
...
...
*/
#pragma trahrhe collapse(2)
for(i = 0; i < N - 1; i++) {
    for(j = i + 1; j < N; j++) {
        for(k = 0; k < N; k++) {
            A[i][j] += B[k][i] * C[k][j];
        }
        A[j][i] = A[i][j];
    }
}
#pragma endtrahrhe
/*
...
...
*/
```

Et `output.c` serait

```
/*
...
...
*/
unsigned int pc;
unsigned upper_bound = i_Ehrhart(N);
unsigned int first_iteration = 1;
#pragma omp parallel for private(i, j, k) firstprivate(first_iteration) schedule(static)
for(pc = 1; pc <= upper_bound; pc++) {
    if(first_iteration) {
        i = i_trahrhe(pc, N);
        j = j_trahrhe(pc, N, i);
        first_iteration = 0;
```

```
    }
    for(k = 0; k < N; k++) {
        A[i][j] += B[k][i] * C[k][j];
    }
    A[j][i] = A[i][j];
    j++;
    if(j >= N) {
        i++;
        j = i + 1;
    }
}
/*
...
...
*/
```

1.2 Todo list (non-exhaustive)

Pré-requis: installation et familiarisation avec les librairies requises.

- [x] Extraire le code entre pragma
- [x] Parser le pragma et récupérer l'argument
- [x] Appeler clan sur ce code
  - [x] Ecrire le code dans un fichier temporaire avec les pragma scop?
- [x] Récupérer les bornes depuis la représentation openscop
- [x] Appeler trahrhe
  - [x] Ecrire le domaine en syntaxe isl `[N] -> { [i, j, k] : 0 < i < N - 1 and i + 1 <= j < N and 0 < k < N }`
  - [x] Récupérer le header c généré
  - [x] Ajouter l'include
- [x] Génération de code
  - [x] Combinaison de cloog et de generation à la main
  - [x] Penser au pragma omp
  - [x] Besoin de connaitre les bornes et les dépendances des itérateurs

Chapter 2

Dependencies

- GMP
- NTL
- Polylib
- isl
- ntl
- Barvinok
- trahrhe
- osl
- clan
- cloog

To install Barvinok, you need to follow these steps:

1. Get GMP using the procedure described in https://libntl.org/doc/tour-gmp.html (note the $HOME/sw)
2. Get NTL using the info in this same page
3. Get ISL
4. You should now be able to get Barvinok
```
. ./configure --with-isl=bundled
make
make install
```

You may need to create a symbolic link to the library in the /usr/lib directory. You can do this by running the following command:
```
sudo ln -s /usr/local/lib/libbarvinok.so.0 /usr/lib/libbarvinok.so.0
```

2.1 ## Troubleshooting

If you are having trouble with the installation of the library, you can try the following commands to copy the library to the /usr/lib directory (assuming you have the library in /usr/local/lib directory):
```
sudo cp /usr/local/lib/lib<LIB_NAME>* /usr/lib
```
If your machine is missing autoreconf command, you can install it by running the following command:
```
apt-get install dh-autoreconf
```

Chapter 3

LICENCE

TDB

Chapter 4

Usage

This section provides a guide on how to use the Automatic Loop Collapsing tool.

4.1 Prerequisites

Before using the Automatic Loop Collapsing tool, ensure that you have the following prerequisites:

- Installed all the dependencies as described in the installation guide.
- Familiarized yourself with the feature definitions.
- Ensure that your input file is in the correct format i.e., it should contain the loops that you want to collapse along with the `#pragma trahrhe collapse(N)` directive.

4.2 Steps

To use the Automatic Loop Collapsing tool, follow these steps:

1. Clone the repository:

```
git clone
```

1. Navigate to the `Automatic-loop-collapsing` directory:
2. Run the following command:

```
(collapse) [input.c] -o [output.c]
```

You can also specify additional options that are:

- `-h` or `--help`: Display the help message.
- `-v` or `--version`: Display the version of the tool.
- `-o` or `--output`: Specify the output file.

The tool will automatically collapse the loops in the input file and generate the output file.

4.3 Example

Consider the following example:

```
#pragma trahrhe collapse(2)
for(i = 0; i < N - 1; i++) {
    for(j = i + 1; j < N; j++) {
        for(k = 0; k < N; k++) {
            A[i][j] += B[k][i] * C[k][j];
        }
        A[j][i] = A[i][j];
    }
}
#pragma endtrahrhe
```

After running the tool, the output file would be:

```
unsigned int pc_0;
unsigned upper_bound_0 = i_Ehrhart0(N);
unsigned int first_iteration_0 = 1;
#pragma omp parallel for private(i, j, k) firstprivate(first_iteration_0) schedule(static)
for(pc_0 = 1; pc_0 <= upper_bound_0; pc_0++) {
    if(first_iteration_0) {
        i = i_trahrhe0(pc_0, N);
        j = j_trahrhe0(pc_0, N, i);
        first_iteration_0 = 0;
    }
    for(k = 0; k < N; k++) {
        A[i][j] += B[k][i] * C[k][j];
    }
    A[j][i] = A[i][j];
    j++;
    if(j >= N) {
        i++;
        j = i + 1;
    }
}
```

Chapter 5

Automatic-loop-collapsing

5.1 Installation

Refer to the installation guide.

5.2 Feature Definitions

Refer to the feature definitions.

5.3 Usage

Refer to the usage guide.

5.4 License

TBD

Chapter 6

Data Structure Index

6.1 Data Structures

Here are the data structures with brief descriptions:

boundary
: Boundary list 15

boundaryList 15

iterationDomain
: Iteration domain representation 16

iterationDomainList 16

TCD_FlowData
: Computational data to be transported during the collapsing 16

Chapter 7

File Index

7.1 File List

Here is a list of all documented files with brief descriptions:

include/codegen.h
: This file contains the code generation functions 17

include/data.h
: Data structures and helper functions to structure the collapsing flow 18

include/flow.h
: Data structures that represent the progress of the trahrhe collapsing process 21

include/format_helper.h 22

include/fs.h
: File System 24

src/codegen.c
: Edits an OpenSCoP representation to generate an output code where loops are collapsed . . 25

src/flow.c
: Flow module implementation 26

src/format_helper.c
: This file contains the functions that help to format the output code 27

src/fs.c
: File System 28

Chapter 8

Data Structure Documentation

8.1 boundary Struct Reference

Boundary list.
`#include <data.h>`
Collaboration diagram for boundary:

Data Fields

- TCD_IterationDomainList firstIterDomainOfUnion
  The iteration domain unions to pass to Trhahre.
- char * outerLoopVar
  Outer loop variable.
- char * outerLoopUpperBound
  Outer loop upper bound.
- char * iterationDomainsString
  Iteration domains string.
- char *** iteratorDependenciesArray
  An array of string representing the list of dependencies of the iterators in the same order as the iterators in the domain.
- char ** nameArray
  The array of the iterator names.
- struct boundary * next
  Next loop boundaries.
- int parametersCount
  The number of parameters.

8.1.1 Detailed Description

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

- include/data.h

8.2 boundaryList Struct Reference

Collaboration diagram for boundaryList:

Data Fields

- TCD_Boundary **first**

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

- include/data.h

8.3 iterationDomain Struct Reference

Iteration domain representation.
`#include <data.h>`
Collaboration diagram for iterationDomain:

Data Fields

- char * iterationDomain

  The iteration domain under the ISL format to pass to Trahrhe.
- struct iterationDomain * **next**

8.3.1 Detailed Description

Iteration domain representation.
The documentation for this struct was generated from the following file:

- include/data.h

8.4 iterationDomainList Struct Reference

Collaboration diagram for iterationDomainList:

Data Fields

- TCD_IterationDomain **first**

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

- include/data.h

8.5 TCD_FlowData Struct Reference

Computational data to be transported during the collapsing.
`#include <flow.h>`

Data Fields

- char * entryFile

  Entry file for the next step.
- char * outputFile

  Output file.
- int * collapseParameters

  Collapse parameters.
- osl_scop_p scop

  Pointer on the current polyedral representation of the source code.

8.5.1 Detailed Description

Computational data to be transported during the collapsing.
The documentation for this struct was generated from the following file:

- include/flow.h

Chapter 9

File Documentation

9.1 include/codegen.h File Reference

This file contains the code generation functions.

```
#include <osl/osl.h>
#include <cloog/isl/cloog.h>
#include "flow.h"
#include "data.h"
#include "format_helper.h"
#include "fs.h"
```

Include dependency graph for codegen.h: This graph shows which files directly or indirectly include this file:

Functions

- void generateCode (TCD_BoundaryList boundaryList)

  Computes the new SCoP structure using the scop in global flow and the boundary list.
- void generateHeaderFile (TCD_BoundaryList boundaryList)

  Generates the header file.
- void mergeGeneratedCode ()

  Generates the source file.
- void removeTemporaryFiles ()

  Removes the temporary files.
- CloogState * **cloog_isl_state_malloc** (struct isl_ctx *ctx)

9.1.1 Detailed Description

This file contains the code generation functions.

Author

SORGHO Nongma

Version

0.1

Date

2024-02-09

Copyright

9.1.2 Function Documentation

9.1.2.1 generateCode()

```
void generateCode (
            TCD_BoundaryList boundaryList )
```

Computes the new SCoP structure using the scop in global flow and the boundary list.

Parameters

boundaryList	The boundary list

Computes the new SCoP structure using the scop in global flow and the boundary list.

Parameters

boundaryList	

9.1.2.2 generateHeaderFile()

```
void generateHeaderFile (
            TCD_BoundaryList boundaryList )
```

Generates the header file.

Parameters

boundaryList	The boundary list

Generates the header file.

Parameters

boundaryList	

9.1.2.3 mergeGeneratedCode()

```
void mergeGeneratedCode ( )
```

Generates the source file.

Parameters

boundaryList	The boundary list

Generates the source file.
Uses a shell script

9.2 include/data.h File Reference

Data structures and helper functions to structure the collapsing flow.

```
#include <stdlib.h>
#include <string.h>
```

```
#include <stdio.h>
#include <ctype.h>
#include "flow.h"
#include "format_helper.h"
```

Include dependency graph for data.h: This graph shows which files directly or indirectly include this file:

Data Structures

- struct iterationDomain

  Iteration domain representation.
- struct iterationDomainList
- struct boundary

  Boundary list.
- struct boundaryList

Typedefs

- typedef struct iterationDomain * **TCD_IterationDomain**
- typedef struct iterationDomainList * **TCD_IterationDomainList**
- typedef struct boundary * **TCD_Boundary**
- typedef struct boundaryList * **TCD_BoundaryList**

Functions

- TCD_IterationDomain copyIterationDomain (TCD_IterationDomain original)

  Copy an iteration domain.
- TCD_Boundary getBoundary (osl_statement_p statement, osl_names_p iteratorStrings)

  Get a boundary given a domain.
- TCD_BoundaryList getBoundaries ()

  Get the Boundaries object from the current scop.
- void printBoundaries (TCD_BoundaryList boundaryList)

  Print the boundaries.
- TCD_Boundary copyBoundary (TCD_Boundary original)

  Copy a boundary.

9.2.1 Detailed Description

Data structures and helper functions to structure the collapsing flow.

Author

Nongma SORGHO

Version

0.1

Date

2024-02-04

9.2.2 Function Documentation

9.2.2.1 copyBoundary()

```
TCD_Boundary copyBoundary (
            TCD_Boundary original )
```

Copy a boundary.

Parameters

original	

Returns

TCD_Boundary

Copy a boundary.

Parameters

original	

Returns

TCD_Boundary

9.2.2.2 copyIterationDomain()

```
TCD_IterationDomain copyIterationDomain (
            TCD_IterationDomain original )
```

Copy an iteration domain.

Parameters

original	

Returns

TCD_IterationDomain

9.2.2.3 getBoundaries()

```
TCD_BoundaryList getBoundaries ( )
```

Get the Boundaries object from the current scop.

Returns

TCD_BoundaryList

Get the Boundaries object from the current scop.

Returns

TCD_BoundaryList

9.2.2.4 getBoundary()

```
TCD_Boundary getBoundary (
            osl_statement_p statement,
            osl_names_p names )
```

Get a boundary given a domain.

Parameters

statement	
iteratorStrings	

Returns

TCD_Boundary

Get a boundary given a domain.

Parameters

statement	
names	

Returns

9.2.2.5 printBoundaries()

```
void printBoundaries (
            TCD_BoundaryList boundaryList )
```

Print the boundaries.

Parameters

boundaryList	Print the boundaries.
boundaryList	

9.3 include/flow.h File Reference

Data structures that represent the progress of the trahrhe collapsing process.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <osl/osl.h>
```

Include dependency graph for flow.h: This graph shows which files directly or indirectly include this file:

Data Structures

- struct TCD_FlowData

  Computational data to be transported during the collapsing.

Macros

- #define **SCOPED_FILENAME** "scope.source.scop"
- #define **COLLAPSE_PARAMETERS_FILENAME** "collapse_parameter.source.txt"
- #define **INTERMEDIATE_FILENAME** "intermediate.source"

Functions

- void initTcdFlow (char ∗inputFilename, char ∗outputFilename)

Inits the Tcd_Flow structure.

- void endTcdFlow ()

  Destruct the Tcd_Flow and frees all memories spaces linked to it.

9.3.1 Detailed Description

Data structures that represent the progress of the trahrhe collapsing process.

Author

Nongma SORGHO

Version

0.1

Date

2024-02-03

9.3.2 Function Documentation

9.3.2.1 endTcdFlow()

```
void endTcdFlow (
            void  )
```

Destruct the Tcd_Flow and frees all memories spaces linked to it.
Destruct the Tcd_Flow and frees all memories spaces linked to it.
Ends the TCD_FlowData structure by freeing the memory allocated

9.3.2.2 initTcdFlow()

```
void initTcdFlow (
            char * inputFilename,
            char * outputFilename )
```

Inits the Tcd_Flow structure.
Inits the Tcd_Flow structure.

Parameters

inputFilename	
outputFilename	

9.4 include/format_helper.h File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

Include dependency graph for format_helper.h: This graph shows which files directly or indirectly include this file:

Functions

- void tabString (FILE *file, char *string, long fsize)

  Adds a tabulation to the beginning of each line of a string and writes it to a file.

- char * tabStringReturn (char *string, long fsize)

Adds a tabulation to the beginning of each line of a string and returns the new string.

- int digit_check (char key[ ])

  Says if a string is a digit.

- char * take (int index, char *string)

  Returns the nth token of a string.

9.4.1 Detailed Description

Author

Nongma SORGHO

Version

0.1

Date

2024-02-13

9.4.2 Function Documentation

9.4.2.1 digit_check()

```
int digit_check (
            char key[] )
```

Says if a string is a digit.

Parameters

key	

Returns

int

9.4.2.2 tabString()

```
void tabString (
            FILE * file,
            char * string,
            long fsize )
```

Adds a tabulation to the beginning of each line of a string and writes it to a file.

Parameters

file	
string	
fsize	

9.4.2.3 tabStringReturn()

```
char* tabStringReturn (
            char * string,
            long fsize )
```

Adds a tabulation to the beginning of each line of a string and returns the new string.

Parameters

index	
string	

Returns

char*

9.4.2.4 take()

```
char* take (
            int index,
            char * string )
```

Returns the nth token of a string.

Parameters

index	
string	

Returns

char*

9.5 include/fs.h File Reference

File System.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
```

Include dependency graph for fs.h: This graph shows which files directly or indirectly include this file:

Functions

- void fs_open (char *filename)

  Open file for writing.
- void fs_close ()

  Close file.
- void fs_writef (char *str,...)

  Write string to file with new line with format.
- void fs_writeft (char *str,...)

  Write string to file with new line and tabular with format.
- char * fs_rewind ()
- void fs_tabular ()

  Write a tabular to file before each line till the opposite is called.
- void fs_untabular ()

Write a tabular to file before each line till the opposite is called.

- void **fs_writefl** (char *str,...)

9.5.1 Detailed Description

File System.

9.5.2 Function Documentation

9.5.2.1 fs_open()

```
void fs_open (
            char * filename )
```

Open file for writing.

Parameters

filename	

9.5.2.2 fs_rewind()

```
char* fs_rewind ( )
```

reads the file's content and positions the stream at the beginning of the file returns the content file inside a buffer

9.5.2.3 fs_writef()

```
void fs_writef (
            char * str,
             ...  )
```

Write string to file with new line with format.

Parameters

str	

9.5.2.4 fs_writeft()

```
void fs_writeft (
            char * str,
             ...  )
```

Write string to file with new line and tabular with format.

Parameters

str	

9.6 src/codegen.c File Reference

Edits an OpenSCoP representation to generate an output code where loops are collapsed.

```
#include "codegen.h"
```

Include dependency graph for codegen.c:

9.7 src/flow.c File Reference

Flow module implementation.
`#include "flow.h"`
Include dependency graph for flow.c:

Functions

- void initTcdFlow (char ∗inputFilename, char ∗outputFilename)

  Initializes the TCD_FlowData structure.
- void endTcdFlow (void)

  Ends the TCD_FlowData structure.

Variables

- TCD_FlowData ∗ tcdFlowData

  TCD_Flow is global - all C sources that use it must declare it as extern.

9.7.1 Detailed Description

Flow module implementation.

Author

Nongma SORGHO

Version

0.1

Date

2024-02-03

9.7.2 Function Documentation

9.7.2.1 endTcdFlow()

```
void endTcdFlow (
            void  )
```

Ends the TCD_FlowData structure.
Destruct the Tcd_Flow and frees all memories spaces linked to it.
Ends the TCD_FlowData structure by freeing the memory allocated

9.7.2.2 initTcdFlow()

```
void initTcdFlow (
            char * inputFilename,
            char * outputFilename )
```

Initializes the TCD_FlowData structure.
Inits the Tcd_Flow structure.

Parameters

inputFilename	
outputFilename	

9.8 src/format_helper.c File Reference

This file contains the functions that help to format the output code.
`#include "format_helper.h"`
Include dependency graph for format_helper.c:

Functions

- void tabString (FILE *file, char *string, long fsize)

  Adds a tabulation to the beginning of each line of a string and writes it to a file.
- char * take (int index, char *string)

  Returns the nth token of a string.
- char * tabStringReturn (char *string, long fsize)

  Adds a tabulation to the beginning of each line of a string and returns the new string.
- int digit_check (char key[ ])

  Says if a string is a digit.

9.8.1 Detailed Description

This file contains the functions that help to format the output code.

Version

0.1

Date

2024-02-13

9.8.2 Function Documentation

9.8.2.1 digit_check()

```
int digit_check (
            char key[] )
```

Says if a string is a digit.

Parameters

key	

Returns

int

9.8.2.2 tabString()

```
void tabString (
            FILE * file,
            char * string,
            long fsize )
```

Adds a tabulation to the beginning of each line of a string and writes it to a file.

Parameters

file	
string	
fsize	

9.8.2.3 tabStringReturn()

```
char* tabStringReturn (
            char * string,
            long fsize )
```

Adds a tabulation to the beginning of each line of a string and returns the new string.

Parameters

index	
string	

Returns

char*

9.8.2.4 take()

```
char* take (
            int index,
            char * string )
```

Returns the nth token of a string.

Parameters

index	
string	

Returns

char*

9.9 src/fs.c File Reference

File System.
`#include "fs.h"`
Include dependency graph for fs.c:

Functions

- void fs_open (char *filename)

  Open file for writing.
- void fs_close ()

  Close file.
- char * fs_rewind ()
- void fs_writef (char *str,...)

Write string to file with new line with format.

- void **fs_writefl** (char *str,...)
- void fs_writeft (char *str,...)

  Write string to file with new line and tabular with format.
- void fs_tabular ()

  Write a tabular to file before each line till the opposite is called.
- void fs_untabular ()

  Write a tabular to file before each line till the opposite is called.

Variables

- FILE * **fs**
- int **tabular** = 0
- char * **outputname**

9.9.1 Detailed Description

File System.

9.9.2 Function Documentation

9.9.2.1 fs_open()

```
void fs_open (
            char * filename )
```

Open file for writing.

Parameters

filename	

9.9.2.2 fs_rewind()

```
char* fs_rewind ( )
```

reads the file's content and positions the stream at the beginning of the file returns the content file inside a buffer

9.9.2.3 fs_writef()

```
void fs_writef (
            char * str,
             ...  )
```

Write string to file with new line with format.

Parameters

str	

9.9.2.4 fs_writeft()

```
void fs_writeft (
            char * str,
```

. . .)

Write string to file with new line and tabular with format.

Parameters

str	

Index

boundary, 15
boundaryList, 15

codegen.h
    generateCode, 18
    generateHeaderFile, 18
    mergeGeneratedCode, 18
copyBoundary
    data.h, 19
copyIterationDomain
    data.h, 20

data.h
    copyBoundary, 19
    copyIterationDomain, 20
    getBoundaries, 20
    getBoundary, 20
    printBoundaries, 21
digit_check
    format_helper.c, 27
    format_helper.h, 23

endTcdFlow
    flow.c, 26
    flow.h, 22

flow.c
    endTcdFlow, 26
    initTcdFlow, 26
flow.h
    endTcdFlow, 22
    initTcdFlow, 22
format_helper.c
    digit_check, 27
    tabString, 27
    tabStringReturn, 28
    take, 28
format_helper.h
    digit_check, 23
    tabString, 23
    tabStringReturn, 23
    take, 24
fs.c
    fs_open, 29
    fs_rewind, 29
    fs_writef, 29
    fs_writeft, 29
fs.h
    fs_open, 25
    fs_rewind, 25
    fs_writef, 25
    fs_writeft, 25
fs_open
    fs.c, 29
    fs.h, 25
fs_rewind
    fs.c, 29
    fs.h, 25
fs_writef
    fs.c, 29
    fs.h, 25
fs_writeft
    fs.c, 29
    fs.h, 25

generateCode
    codegen.h, 18
generateHeaderFile
    codegen.h, 18
getBoundaries
    data.h, 20
getBoundary
    data.h, 20

include/codegen.h, 17
include/data.h, 18
include/flow.h, 21
include/format_helper.h, 22
include/fs.h, 24
initTcdFlow
    flow.c, 26
    flow.h, 22
iterationDomain, 16
iterationDomainList, 16

mergeGeneratedCode
    codegen.h, 18

printBoundaries
    data.h, 21

src/codegen.c, 25
src/flow.c, 26
src/format_helper.c, 27
src/fs.c, 28

tabString
    format_helper.c, 27
    format_helper.h, 23
tabStringReturn
    format_helper.c, 28

format_helper.h, 23
take
format_helper.c, 28
format_helper.h, 24
TCD_FlowData, 16