

SS_PROJEKAT

Generated by Doxygen 1.8.20

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 equNode Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Data Documentation	5
3.1.2.1 equation	5
3.1.2.2 s	5
3.1.2.3 section	6
3.1.2.4 symbol	6
3.2 Flink_Node Struct Reference	6
3.2.1 Detailed Description	6
3.2.2 Member Data Documentation	6
3.2.2.1 adress	6
3.2.2.2 op	6
3.3 pomNode Struct Reference	7
3.3.1 Detailed Description	7
3.3.2 Constructor & Destructor Documentation	7
3.3.2.1 pomNode()	7
3.3.3 Member Data Documentation	7
3.3.3.1 locationCounter_	7
3.3.3.2 nextLC	8
3.3.3.3 section_	8
3.3.3.4 symbol_	8
3.4 relNode Struct Reference	8
3.4.1 Detailed Description	8
3.4.2 Member Data Documentation	8
3.4.2.1 adress	9
3.4.2.2 offset	9
3.4.2.3 section	9
3.4.2.4 symbol	9
3.4.2.5 type	9
3.5 sectionNode Struct Reference	9
3.5.1 Detailed Description	10
3.5.2 Member Data Documentation	10
3.5.2.1 GL	10
3.5.2.2 name	10
3.5.2.3 offset	10
3.5.2.4 realLC	10

3.5.2.5 rwa	10
3.6 symbolNode Struct Reference	10
3.6.1 Detailed Description	11
3.6.2 Constructor & Destructor Documentation	11
3.6.2.1 symbolNode()	11
3.6.3 Member Data Documentation	11
3.6.3.1 defined	11
3.6.3.2 equSection	11
3.6.3.3 Flink	12
3.6.3.4 isEqu	12
3.6.3.5 LG	12
3.6.3.6 name	12
3.6.3.7 section	12
3.6.3.8 size	12
3.6.3.9 value	12
4 File Documentation	13
4.1 C:/Users/DUSAN/Desktop/ss PROJEKAT/Adressing.cpp File Reference	13
4.1.1 Function Documentation	14
4.1.1.1 determinateAdressing()	14
4.1.1.2 immed()	14
4.1.1.3 memdir()	14
4.1.1.4 regdir()	14
4.1.1.5 regind()	14
4.1.1.6 regindpom()	15
4.1.2 Variable Documentation	15
4.1.2.1 sym	15
4.2 C:/Users/DUSAN/Desktop/ss PROJEKAT/Adressing.h File Reference	15
4.2.1 Function Documentation	15
4.2.1.1 determinateAdressing()	16
4.2.1.2 immed()	16
4.2.1.3 memdir()	16
4.2.1.4 regdir()	16
4.2.1.5 regind()	16
4.2.1.6 regindpom()	17
4.2.2 Variable Documentation	17
4.2.2.1 sym	17
4.3 C:/Users/DUSAN/Desktop/ss PROJEKAT/DigitConvert.cpp File Reference	17
4.3.1 Function Documentation	17
4.3.1.1 intToHex()	17
4.3.1.2 makeTwoBytes()	18
4.3.1.3 makeWord()	18

4.4 C:/Users/DUSAN/Desktop/ss PROJEKAT/DigitConvert.h File Reference	18
4.4.1 Function Documentation	18
4.4.1.1 intToHex()	18
4.4.1.2 makeTwoBytes()	19
4.4.1.3 makeWord()	19
4.5 C:/Users/DUSAN/Desktop/ss PROJEKAT/FlinkStruct.h File Reference	19
4.6 C:/Users/DUSAN/Desktop/ss PROJEKAT/main.cpp File Reference	19
4.6.1 Function Documentation	20
4.6.1.1 main()	20
4.7 C:/Users/DUSAN/Desktop/ss PROJEKAT/outputData.cpp File Reference	20
4.7.1 Function Documentation	20
4.7.1.1 writeData()	20
4.7.1.2 writeErrorDest()	21
4.7.1.3 writeErrorNotDefined()	21
4.7.1.4 writeExternError()	21
4.8 C:/Users/DUSAN/Desktop/ss PROJEKAT/outputData.h File Reference	21
4.8.1 Function Documentation	21
4.8.1.1 writeData()	21
4.8.1.2 writeErrorDest()	22
4.8.1.3 writeErrorNotDefined()	22
4.8.1.4 writeExternError()	22
4.9 C:/Users/DUSAN/Desktop/ss PROJEKAT/RegexSearch.cpp File Reference	22
4.9.1 Function Documentation	23
4.9.1.1 adressing()	23
4.9.1.2 changeArg()	24
4.9.1.3 countAllEques()	24
4.9.1.4 countEqu()	24
4.9.1.5 makeFirstByte()	24
4.9.1.6 makeSecondByte()	24
4.9.1.7 putInEquList()	25
4.9.1.8 searchEG()	25
4.9.1.9 searchEqu()	25
4.9.1.10 searchIns()	25
4.9.1.11 searchIns2()	25
4.9.1.12 searchInsNoArg()	26
4.9.1.13 searchLabel()	26
4.9.1.14 searchSection()	26
4.9.1.15 searchSWB()	26
4.9.2 Variable Documentation	26
4.9.2.1 equList	26
4.9.2.2 firstArg	27
4.9.2.3 inImmed	27

4.9.2.4	jump	27
4.9.2.5	lowHigh	27
4.9.2.6	oneArg	27
4.10	C:/Users/DUSAN/Desktop/ss PROJEKAT/RegexSearch.h File Reference	27
4.10.1	Function Documentation	28
4.10.1.1	adressing()	29
4.10.1.2	changeArg()	29
4.10.1.3	countAllEques()	29
4.10.1.4	countEqu()	29
4.10.1.5	makeFirstByte()	29
4.10.1.6	makeSecondByte()	30
4.10.1.7	putInEquList()	30
4.10.1.8	searchEG()	30
4.10.1.9	searchEqu()	30
4.10.1.10	searchIns()	30
4.10.1.11	searchIns2()	31
4.10.1.12	searchInsNoArg()	31
4.10.1.13	searchLabel()	31
4.10.1.14	searchSection()	31
4.10.1.15	searchSWB()	31
4.10.2	Variable Documentation	32
4.10.2.1	equList	32
4.10.2.2	firstArg	32
4.10.2.3	inImmed	32
4.10.2.4	jump	32
4.10.2.5	lowHigh	32
4.10.2.6	oneArg	32
4.11	C:/Users/DUSAN/Desktop/ss PROJEKAT/Relocation.cpp File Reference	33
4.11.1	Function Documentation	33
4.11.1.1	addList()	33
4.11.1.2	addToPomPcList()	34
4.11.1.3	addToRelTable()	34
4.11.1.4	fillPomBytes()	34
4.11.1.5	fillRelTable()	34
4.11.2	Variable Documentation	34
4.11.2.1	nextLC	34
4.11.2.2	pomPCList	35
4.11.2.3	relTables	35
4.12	C:/Users/DUSAN/Desktop/ss PROJEKAT/Relocation.h File Reference	35
4.12.1	Function Documentation	36
4.12.1.1	addList()	36
4.12.1.2	addToPomPcList()	36

4.12.1.3 addToRelTable()	36
4.12.1.4 fillPomBytes()	36
4.12.1.5 fillRelTable()	36
4.12.2 Variable Documentation	37
4.12.2.1 nextLC	37
4.12.2.2 pomPCList	37
4.12.2.3 relTables	37
4.13 C:/Users/DUSAN/Desktop/ss PROJEKAT/SectionTable.cpp File Reference	37
4.13.1 Function Documentation	37
4.13.1.1 findLC()	38
4.13.1.2 setSection()	38
4.13.2 Variable Documentation	38
4.13.2.1 sectionTable	38
4.14 C:/Users/DUSAN/Desktop/ss PROJEKAT/SectionTable.h File Reference	38
4.14.1 Function Documentation	39
4.14.1.1 findLC()	39
4.14.1.2 setSection()	39
4.14.2 Variable Documentation	39
4.14.2.1 sectionTable	39
4.15 C:/Users/DUSAN/Desktop/ss PROJEKAT/SymbolTable.cpp File Reference	39
4.15.1 Function Documentation	40
4.15.1.1 backpatching()	40
4.15.1.2 changeLC()	41
4.15.1.3 findSymbol()	41
4.15.1.4 findSymbolBP()	41
4.15.1.5 findSymbolPC()	41
4.15.1.6 findSymbolREL()	41
4.15.1.7 findSymbolRELOCATION()	42
4.15.1.8 findSymbolSEC()	42
4.15.1.9 setSymbol()	42
4.15.1.10 setSymbolExtern()	42
4.15.1.11 setSymbolGlobal()	42
4.15.1.12 setSymbolLabel()	43
4.15.1.13 writeList()	43
4.15.2 Variable Documentation	43
4.15.2.1 equ	43
4.15.2.2 equSearched	43
4.15.2.3 locationCounter	43
4.15.2.4 opCode	43
4.15.2.5 section	44
4.15.2.6 symbol_Table	44
4.16 C:/Users/DUSAN/Desktop/ss PROJEKAT/SymbolTable.h File Reference	44

4.16.1 Function Documentation	45
4.16.1.1 backpatching()	45
4.16.1.2 changeLC()	45
4.16.1.3 findSymbol()	45
4.16.1.4 findSymbolBP()	45
4.16.1.5 findSymbolPC()	46
4.16.1.6 findSymbolREL()	46
4.16.1.7 findSymbolRELOCATION()	46
4.16.1.8 findSymbolSEC()	46
4.16.1.9 setSymbol()	47
4.16.1.10 setSymbolExtern()	47
4.16.1.11 setSymbolGlobal()	47
4.16.1.12 setSymbolLabel()	47
4.16.1.13 writeList()	47
4.16.2 Variable Documentation	48
4.16.2.1 equ	48
4.16.2.2 equSearched	48
4.16.2.3 locationCounter	48
4.16.2.4 opCode	48
4.16.2.5 section	48
4.16.2.6 symbol_Table	48
Index	49

Chapter 1

Class Index

1.1 Class List

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

equNode	
Cvor u listi koja služi da računava .equ izraze	5
Flink_Node	
Jedan cvor u listi Flink(lista koja pamti gdje se pojavljuje simbol)	6
pomNode	
Cvor u listi koja služi da računava .equ izraze	7
relNode	
Cvor u listi koja čuva relokacione zapise	8
sectionNode	
Cvor u listi za sekcije	9
symbolNode	
Cvor u listi rabele simbola	10

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/DUSAN/Desktop/ss PROJEKAT/ Adressing.cpp	13
C:/Users/DUSAN/Desktop/ss PROJEKAT/ Adressing.h	15
C:/Users/DUSAN/Desktop/ss PROJEKAT/ DigitConvert.cpp	17
C:/Users/DUSAN/Desktop/ss PROJEKAT/ DigitConvert.h	18
C:/Users/DUSAN/Desktop/ss PROJEKAT/ FlinkStruct.h	19
C:/Users/DUSAN/Desktop/ss PROJEKAT/ main.cpp	19
C:/Users/DUSAN/Desktop/ss PROJEKAT/ outputData.cpp	20
C:/Users/DUSAN/Desktop/ss PROJEKAT/ outputData.h	21
C:/Users/DUSAN/Desktop/ss PROJEKAT/ RegexSearch.cpp	22
C:/Users/DUSAN/Desktop/ss PROJEKAT/ RegexSearch.h	27
C:/Users/DUSAN/Desktop/ss PROJEKAT/ Relocation.cpp	33
C:/Users/DUSAN/Desktop/ss PROJEKAT/ Relocation.h	35
C:/Users/DUSAN/Desktop/ss PROJEKAT/ SectionTable.cpp	37
C:/Users/DUSAN/Desktop/ss PROJEKAT/ SectionTable.h	38
C:/Users/DUSAN/Desktop/ss PROJEKAT/ SymbolTable.cpp	39
C:/Users/DUSAN/Desktop/ss PROJEKAT/ SymbolTable.h	44

Chapter 3

Class Documentation

3.1 equNode Struct Reference

Cvor u listi koja služi da računa .equ izraze.

```
#include <RegexSearch.h>
```

Public Attributes

- string **symbol**
- string **equation**
- int **section**
- int **s**

3.1.1 Detailed Description

Cvor u listi koja služi da računa .equ izraze.

3.1.2 Member Data Documentation

3.1.2.1 equation

```
string equNode::equation
```

3.1.2.2 s

```
int equNode::s
```

3.1.2.3 section

```
int equNode::section
```

3.1.2.4 symbol

```
string equNode::symbol
```

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

- C:/Users/DUSAN/Desktop/ss PROJEKAT/ **RegexSearch.h**

3.2 Flink_Node Struct Reference

Jedan cvor u listi Flink(lista koja pamti gde se pojavljuje simbol)

```
#include <FlinkStruct.h>
```

Public Attributes

- int **adress**
- char **op**

3.2.1 Detailed Description

Jedan cvor u listi Flink(lista koja pamti gde se pojavljuje simbol)

3.2.2 Member Data Documentation

3.2.2.1 adress

```
int Flink_Node::adress
```

3.2.2.2 op

```
char Flink_Node::op
```

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

- C:/Users/DUSAN/Desktop/ss PROJEKAT/ **FlinkStruct.h**

3.3 pomNode Struct Reference

Cvor u listi koja služi da računa .equ izraze.

```
#include <Relocation.h>
```

Public Member Functions

- **pomNode** (int lc, int next, string symbol, int **section**)

Public Attributes

- int **locationCounter_**
- int **nextLC**
- string **symbol_**
- int **section_**

3.3.1 Detailed Description

Cvor u listi koja služi da računa .equ izraze.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 pomNode()

```
pomNode::pomNode (
    int lc,
    int next,
    string symbol,
    int section ) [inline]
```

3.3.3 Member Data Documentation

3.3.3.1 locationCounter_

```
int pomNode::locationCounter_
```

3.3.3.2 nextLC

```
int pomNode::nextLC
```

3.3.3.3 section_

```
int pomNode::section_
```

3.3.3.4 symbol_

```
string pomNode::symbol_
```

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

- C:/Users/DUSAN/Desktop/ss PROJEKAT/ **Relocation.h**

3.4 relNode Struct Reference

Cvor u listi koja cuva relokacione zapise.

```
#include <Relocation.h>
```

Public Attributes

- string **offset**
- string **type**
- string **symbol**
- int **section**
- int **adressing**

3.4.1 Detailed Description

Cvor u listi koja cuva relokacione zapise.

3.4.2 Member Data Documentation

3.4.2.1 addressing

```
int relNode::addressing
```

3.4.2.2 offset

```
string relNode::offset
```

3.4.2.3 section

```
int relNode::section
```

3.4.2.4 symbol

```
string relNode::symbol
```

3.4.2.5 type

```
string relNode::type
```

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

- C:/Users/DUSAN/Desktop/ss PROJEKAT/ **Relocation.h**

3.5 sectionNode Struct Reference

Cvor u listi za sekcije.

```
#include <SectionTable.h>
```

Public Attributes

- string **name**
- string **offset**
- string **rwa**
- char **GL**
- int **realLC**

3.5.1 Detailed Description

Cvor u listi za sekcije.

3.5.2 Member Data Documentation

3.5.2.1 GL

```
char sectionNode::GL
```

3.5.2.2 name

```
string sectionNode::name
```

3.5.2.3 offset

```
string sectionNode::offset
```

3.5.2.4 realLC

```
int sectionNode::realLC
```

3.5.2.5 rwa

```
string sectionNode::rwa
```

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

- C:/Users/DUSAN/Desktop/ss PROJEKAT/ **SectionTable.h**

3.6 symbolNode Struct Reference

Covr u listi rabele simbola.

```
#include <SymbolTable.h>
```

Public Member Functions

- **symbolNode** (string name_, int value_)

Public Attributes

- string **name**
- int **value**
- int **size**
- bool **defined**
- char **LG**
- int **section**
- bool **isEqu**
- int **equSection**
- list< **Flink_Node** > **Flink**

3.6.1 Detailed Description

Covr u listi rabele simbola.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 symbolNode()

```
symbolNode::symbolNode (  
    string name_,  
    int value_ ) [inline]
```

3.6.3 Member Data Documentation

3.6.3.1 defined

```
bool symbolNode::defined
```

3.6.3.2 equSection

```
int symbolNode::equSection
```

3.6.3.3 Flink

```
list< Flink_Node> symbolNode::Flink
```

3.6.3.4 isEqu

```
bool symbolNode::isEqu
```

3.6.3.5 LG

```
char symbolNode::LG
```

3.6.3.6 name

```
string symbolNode::name
```

3.6.3.7 section

```
int symbolNode::section
```

3.6.3.8 size

```
int symbolNode::size
```

3.6.3.9 value

```
int symbolNode::value
```

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

- C:/Users/DUSAN/Desktop/ss PROJEKAT/ **SymbolTable.h**

Chapter 4

File Documentation

4.1 C:/Users/DUSAN/Desktop/ss PROJEKAT/Adressing.cpp File Reference

```
#include <iostream>
#include "Adressing.h"
#include <string>
#include <regex>
#include "DigitConvert.h"
#include "SymbolTable.h"
#include "RegexSearch.h"
#include "Relocation.h"
```

Functions

- int **regdir** (string arg1)
Ispituje da li je adresiranje registarsko direktno.
- int **regind** (string arg1)
Ispituje da li je adresiranje registarsko indirektno.
- int **regindpom** (string arg1, int &pom)
Ispituje da li je adresiranje registarsko indirektno sa pomerajem.
- int **memdir** (string arg)
Ispituje da li je adresiranje memorijsko direktno.
- int **immed** (string arg1)
Ispituje da li je adresiranje neposredno.
- int **determinateAdressing** (string arg, int &pom, int &adressing)
Fja koja ispituje koje je adresiranje.

Variables

- bool **sym** = false

4.1.1 Function Documentation

4.1.1.1 determinateAdressing()

```
int determinateAdressing (
    string arg,
    int & pom,
    int & adresssing )
```

Fja koja ispituje koje je adresiranje.

4.1.1.2 immed()

```
int immed (
    string arg1 )
```

Ispituje da li je adresiranje neposredno.

4.1.1.3 memdir()

```
int memdir (
    string arg )
```

Ispituje da li je adresiranje memorijsko direktno.

4.1.1.4 regdir()

```
int regdir (
    string arg1 )
```

Ispituje da li je adresiranje registarsko direktno.

4.1.1.5 regind()

```
int regind (
    string arg1 )
```

Ispituje da li je adresiranje registarsko indirektno.

4.1.1.6 regindpom()

```
int regindpom (
    string arg1,
    int & pom )
```

Ispituje da li je adresiranje registarsko indirektno sa pomerajem.

4.1.2 Variable Documentation

4.1.2.1 sym

```
bool sym = false
```

4.2 C:/Users/DUSAN/Desktop/ss PROJEKAT/Adressing.h File Reference

```
#include <iostream>
#include <string>
```

Functions

- int **regdir** (string arg1)
Ispituje da li je adresiranje registarsko direktno.
- int **regind** (string arg1)
Ispituje da li je adresiranje registarsko indirektno.
- int **regindpom** (string arg1, int &pom)
Ispituje da li je adresiranje registarsko indirektno sa pomerajem.
- int **memdir** (string arg1)
Ispituje da li je adresiranje memorijsko direktno.
- int **immed** (string arg)
Ispituje da li je adresiranje neposredno.
- int **determinateAdressing** (string arg, int &pom, int & **adressing**)
Fja koja ispituje koje je adresiranje.

Variables

- bool **sym**

4.2.1 Function Documentation

4.2.1.1 determinateAdressing()

```
int determinateAdressing (
    string arg,
    int & pom,
    int & adressing )
```

Fja koja ispituje koje je adresiranje.

4.2.1.2 immed()

```
int immed (
    string arg )
```

Ispituje da li je adresiranje neposredno.

4.2.1.3 memdir()

```
int memdir (
    string arg1 )
```

Ispituje da li je adresiranje memorijsko direktno.

4.2.1.4 regdir()

```
int regdir (
    string arg1 )
```

Ispituje da li je adresiranje registarsko direktno.

4.2.1.5 regind()

```
int regind (
    string arg1 )
```

Ispituje da li je adresiranje registarsko indirektno.

4.2.1.6 regindpom()

```
int regindpom (
    string arg1,
    int & pom )
```

Ispituje da li je adresiranje registarsko indirektno sa pomerajem.

4.2.2 Variable Documentation

4.2.2.1 sym

```
bool sym [extern]
```

4.3 C:/Users/DUSAN/Desktop/ss PROJEKAT/DigitConvert.cpp File Reference

```
#include <iostream>
#include <string>
#include "DigitConvert.h"
#include "SymbolTable.h"
#include <iomanip>
```

Functions

- string **intToHex** (int num)
Funkcija koja konvertuje integer u hexa vrednost.
- string **makeTwoBytes** (string hex)
Pravi dva bajta u hexa predstavu.
- string **makeWord** (string first, string second, string third, string fourth, string fifth, string sixst, string seventh)
Pravi rec za masinski kod u velicini od 3 do 7 bajtova.

4.3.1 Function Documentation

4.3.1.1 intToHex()

```
string intToHex (
    int num )
```

Funkcija koja konvertuje integer u hexa vrednost.

4.3.1.2 makeTwoBytes()

```
string makeTwoBytes (
    string hex )
```

Pravi dva bajta u hexa predstavu.

4.3.1.3 makeWord()

```
string makeWord (
    string first,
    string second,
    string third,
    string fourth,
    string fifth,
    string sixst,
    string seventh )
```

Pravi rec za masinski kod u velicini od 3 do 7 bajtova.

4.4 C:/Users/DUSAN/Desktop/ss PROJEKAT/DigitConvert.h File Reference

```
#include <string>
```

Functions

- string **intToHex** (int num)
Funkcija koja konvertuje integer u hexa vrednost.
- string **makeTwoBytes** (string hex)
Pravi dva bajta u hexa predstavu.
- string **makeWord** (string first, string second, string thrid, string fourth, string fifth, string sixst, string seventh)
Pravi rec za masinski kod u velicini od 3 do 7 bajtova.

4.4.1 Function Documentation

4.4.1.1 intToHex()

```
string intToHex (
    int num )
```

Funkcija koja konvertuje integer u hexa vrednost.

4.4.1.2 makeTwoBytes()

```
string makeTwoBytes (
    string hex )
```

Pravi dva bajta u hexa predstavu.

4.4.1.3 makeWord()

```
string makeWord (
    string first,
    string second,
    string thrid,
    string fourth,
    string fifth,
    string sixst,
    string seventh )
```

Pravi rec za masinski kod u velicini od 3 do 7 bajtova.

4.5 C:/Users/DUSAN/Desktop/ss PROJEKAT/FlinkStruct.h File Reference

Classes

- struct **Flink_Node**

Jedan cvor u listi Flink(lista koja pamti gde se pojavljuje simbol)

4.6 C:/Users/DUSAN/Desktop/ss PROJEKAT/main.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <regex>
#include "RegexSearch.h"
#include "SymbolTable.h"
#include "SectionTable.h"
#include "Relocation.h"
#include "outputData.h"
```

Functions

- int **main** ()

4.6.1 Function Documentation

4.6.1.1 main()

```
int main ( )
```

4.7 C:/Users/DUSAN/Desktop/ss PROJEKAT/outputData.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <ostream>
#include <iomanip>
#include "outputData.h"
#include "SymbolTable.h"
#include "RegexSearch.h"
#include "SectionTable.h"
#include "Relocation.h"
```

Functions

- void **writeData** ()
Ispisuje u izlazni fajl sve podatke.
- void **writeErrorDest** ()
Ispisuje gresku u det operandu.
- void **writeExternError** ()
Ispisuje gresku u extern ako je simbol dva puta definisan.
- void **writeErrorNotDefined** ()
Ispisuje gresku ako simbol nije definisan.

4.7.1 Function Documentation

4.7.1.1 writeData()

```
void writeData ( )
```

Ispisuje u izlazni fajl sve podatke.

4.7.1.2 writeErrorDest()

```
void writeErrorDest ( )
```

Ispisuje gresku u det operandu.

4.7.1.3 writeErrorNotDefined()

```
void writeErrorNotDefined ( )
```

Ispisuje gresku ako simbol nije definisan.

4.7.1.4 writeExternError()

```
void writeExternError ( )
```

Ispisuje gresku u extern ako je simbol dva puta definisan.

4.8 C:/Users/DUSAN/Desktop/ss PROJEKAT/outputData.h File Reference

Functions

- void **writeData** ()
Ispisuje u izlazni fajl sve podatke.
- void **writeErrorDest** ()
Ispisuje gresku u det operandu.
- void **writeExternError** ()
Ispisuje gresku u extern ako je simbol dva puta definisan.
- void **writeErrorNotDefined** ()
Ispisuje gresku ako simbol nije definisan.

4.8.1 Function Documentation

4.8.1.1 writeData()

```
void writeData ( )
```

Ispisuje u izlazni fajl sve podatke.

4.8.1.2 writeErrorDest()

```
void writeErrorDest ( )
```

Ispisuje gresku u det operandu.

4.8.1.3 writeErrorNotDefined()

```
void writeErrorNotDefined ( )
```

Ispisuje gresku ako simbol nije definisan.

4.8.1.4 writeExternError()

```
void writeExternError ( )
```

Ispisuje gresku u extern ako je simbol dva puta definisan.

4.9 C:/Users/DUSAN/Desktop/ss PROJEKAT/RegexSearch.cpp File Reference

```
#include "RegexSearch.h"
#include "SymbolTable.h"
#include <regex>
#include <string>
#include <list>
#include "DigitConvert.h"
#include "Adressing.h"
#include "SectionTable.h"
#include "Relocation.h"
#include "outputData.h"
```

Functions

- bool **searchInsNoArg** (string &line, int &code)
Provera da li je instrukcija bez argumenata.
- bool **searchLabel** (string &line, string &arg, int &code)
Provera da li je labela.
- bool **searchIns** (string &line, string &arg, int &code)
Provera da li je instrukcija sa jednim argumentom.
- bool **searchIns2** (string &line, string &arg1, string &arg2, int &code)
Provera da li je instrukcija sa dva argumenta.
- bool **searchSWB** (string &line)

- Poverava da li je procitana linija .word , .skip ili . byte.*
- bool **searchEG** (string &line)
Ispisuje u .global ili .extern.
- bool **searchEqu** (string &line)
Poverava da li je .equ izraz.
- bool **searchSection** (string &line)
Ispituje da li je sekcija.
- void **adressing** (string line, char &c, string ins)
Proverava koje je adresiranje.
- int **makeFirstByte** (int code, char c)
Pravi prvi bajt za masinski kod.
- string **makeSecondByte** (string arg, char c, string &thirdByte, string &fourthByte)
Pravi bajtove od 3 do 7 za masinski kod.
- bool **countEqu** (string &arg, int &equSection)
Racuna .equ izraz.
- void **putInEquList** (string arg1, string arg2)
Dodaje u listu izraz koji ne moze da se izracuna odmah.
- string **changeArg** (string arg1)
Menja argumen pc u r7 radi lakseg racunanja kasnije.
- void **countAllEques** ()
Racuna sve .equ izraze koje nisu mogli biti izracunati tokom prolaska.

Variables

- list< **equNode** > **equList**
Lista koja cuva .equ izraze.
- bool **jump** = false
- bool **firstArg** = false
- bool **oneArg** = false
- bool **lowHigh** = false
- bool **inImmed** = false

4.9.1 Function Documentation

4.9.1.1 adressing()

```
void adressing (
    string line,
    char & c,
    string ins )
```

Proverava koje je adresiranje.

4.9.1.2 changeArg()

```
string changeArg (
    string arg1 )
```

Menja argumen pc u r7 radi lakseg racunanja kasnije.

4.9.1.3 countAllEques()

```
void countAllEques ( )
```

Racuna sve .equ izraze koje nisu mogli biti izracunati tokom prolaska.

4.9.1.4 countEqu()

```
bool countEqu (
    string & arg,
    int & equSection )
```

Racuna .equ izraz.

4.9.1.5 makeFirstByte()

```
int makeFirstByte (
    int code,
    char c )
```

Pravi prvi bajt za masinski kod.

4.9.1.6 makeSecondByte()

```
string makeSecondByte (
    string arg,
    char c,
    string & thirdByte,
    string & fourthByte )
```

Pravi bajtove od 3 do 7 za masinski kod.

4.9.1.7 putInEquList()

```
void putInEquList (
    string arg1,
    string arg2 )
```

Dodaje u listu izraz koji ne moze da se izracuna odmah.

4.9.1.8 searchEG()

```
bool searchEG (
    string & line )
```

Ispisuje u .global ili .extern.

4.9.1.9 searchEqu()

```
bool searchEqu (
    string & line )
```

Poverava da li je .equ izraz.

4.9.1.10 searchIns()

```
bool searchIns (
    string & line,
    string & arg,
    int & code )
```

Proverava da li je instrukcija sa jednim argumentom.

4.9.1.11 searchIns2()

```
bool searchIns2 (
    string & line,
    string & arg1,
    string & arg2,
    int & code )
```

Proverava da li je instrukcija sa dva argumenta.

4.9.1.12 searchInsNoArg()

```
bool searchInsNoArg (
    string & line,
    int & code )
```

Proverava da li je instrukcija bez argumenata.

4.9.1.13 searchLabel()

```
bool searchLabel (
    string & line,
    string & arg,
    int & code )
```

Poverava da li je labela.

4.9.1.14 searchSection()

```
bool searchSection (
    string & line )
```

Ispituje da li je sekcija.

4.9.1.15 searchSWB()

```
bool searchSWB (
    string & line )
```

Poverava da li je procitana linija .word , .skip ili . byte.

4.9.2 Variable Documentation

4.9.2.1 equList

```
list< equNode> equList
```

Lista koja cuva .equ izraze.

4.9.2.2 firstArg

```
bool firstArg = false
```

4.9.2.3 inImmed

```
bool inImmed = false
```

4.9.2.4 jump

```
bool jump = false
```

4.9.2.5 lowHigh

```
bool lowHigh = false
```

4.9.2.6 oneArg

```
bool oneArg = false
```

4.10 C:/Users/DUSAN/Desktop/ss PROJEKAT/RegexSearch.h File Reference

```
#include <string>
#include <list>
#include <iostream>
```

Classes

- struct **equNode**

Cvor u listi koja služi da racuna .equ izraze.

Functions

- bool **searchInsNoArg** (string &line, int &code)
Proverava da li je instrukcija bez argumenata.
- bool **searchLabel** (string &line, string &arg, int &code)
Poverava da li je labela.
- bool **searchIns** (string &line, string &arg, int &code)
Proverava da li je instrukcija sa jednim argumentom.
- bool **searchIns2** (string &line, string &arg1, string &arg2, int &code)
Proverava da li je instrukcija sa dva argumenta.
- bool **searchEqu** (string &line)
Poverava da li je .equ izraz.
- bool **searchSWB** (string &line)
Poverava da li je procitana linija .word , .skip ili . byte.
- bool **searchEG** (string &line)
Ispisuje u .global ili .extern.
- bool **searchSection** (string &line)
Ispituje da li je sekcija.
- void **adressing** (string line, char &c, string ins)
Proverava koje je adresiranje.
- int **makeFirstByte** (int code, char c)
Pravi prvi bajt za masinski kod.
- string **makeSecondByte** (string arg, char c, string &thirdByte, string &fourthByte)
Pravi bajtove od 3 do 7 za masinski kod.
- bool **countEqu** (string &arg1, int &equSection)
Racuna .equ izraz.
- void **countAllEques** ()
Racuna sve .equ izraze koje nisu mogli biti izracunati tokom prolaska.
- void **putInEquList** (string arg1, string agr2)
Dodaje u listu izraz koji ne moze da se izracuna odmah.
- string **changeArg** (string arg1)
Menja argumen pc u r7 radi lakseg racunanja kasnije.

Variables

- bool **jump**
- bool **firstArg**
- bool **oneArg**
- bool **inImmed**
- bool **lowHigh**
- list< **equNode** > **equList**
Lista koja cuva .equ izraze.

4.10.1 Function Documentation

4.10.1.1 `adressing()`

```
void addressing (
    string line,
    char & c,
    string ins )
```

Proverava koje je adresiranje.

4.10.1.2 `changeArg()`

```
string changeArg (
    string arg1 )
```

Menja argumen pc u r7 radi lakseg racunanja kasnije.

4.10.1.3 `countAllEques()`

```
void countAllEques ( )
```

Racuna sve .equ izraze koje nisu mogli biti izracunati tokom prolaska.

4.10.1.4 `countEqu()`

```
bool countEqu (
    string & arg1,
    int & equSection )
```

Racuna .equ izraz.

4.10.1.5 `makeFirstByte()`

```
int makeFirstByte (
    int code,
    char c )
```

Pravi prvi bajt za masinski kod.

4.10.1.6 makeSecondByte()

```
string makeSecondByte (
    string arg,
    char c,
    string & thirdByte,
    string & fourthByte )
```

Pravi bajtove od 3 do 7 za masinski kod.

4.10.1.7 putInEquList()

```
void putInEquList (
    string arg1,
    string agr2 )
```

Dodaje u listu izraz koji ne moze da se izracuna odmah.

4.10.1.8 searchEG()

```
bool searchEG (
    string & line )
```

Ispisuje u .global ili .extern.

4.10.1.9 searchEqu()

```
bool searchEqu (
    string & line )
```

Poverava da li je .equ izraz.

4.10.1.10 searchIns()

```
bool searchIns (
    string & line,
    string & arg,
    int & code )
```

Proverava da li je instrukcija sa jednim argumentom.

4.10.1.11 searchIns2()

```
bool searchIns2 (
    string & line,
    string & arg1,
    string & arg2,
    int & code )
```

Proverava da li je instrukcija sa dva argumenta.

4.10.1.12 searchInsNoArg()

```
bool searchInsNoArg (
    string & line,
    int & code )
```

Proverava da li je instrukcija bez argumenata.

4.10.1.13 searchLabel()

```
bool searchLabel (
    string & line,
    string & arg,
    int & code )
```

Poverava da li je labela.

4.10.1.14 searchSection()

```
bool searchSection (
    string & line )
```

Ispituje da li je sekcija.

4.10.1.15 searchSWB()

```
bool searchSWB (
    string & line )
```

Poverava da li je procitana linija .word , .skip ili . byte.

4.10.2 Variable Documentation

4.10.2.1 equList

```
list< equNode> equList [extern]
```

Lista koja cuva .equ izraze.

4.10.2.2 firstArg

```
bool firstArg [extern]
```

4.10.2.3 inImmed

```
bool inImmed [extern]
```

4.10.2.4 jump

```
bool jump [extern]
```

4.10.2.5 lowHigh

```
bool lowHigh [extern]
```

4.10.2.6 oneArg

```
bool oneArg [extern]
```


4.11 C:/Users/DUSAN/Desktop/ss PROJEKAT/Relocation.cpp File Reference

```
#include <iostream>
#include <string>
#include <list>
#include "Relocation.h"
#include "SymbolTable.h"
#include "DigitConvert.h"
#include "RegexSearch.h"
#include "SectionTable.h"
```

Functions

- void **addToPomPcList** (int lc, int next, string symbol, int **section**)
Dodaje u listu koja pomaze u racunanju pomeraja za pc rel.
- void **fillPomBytes** ()
Popunjava bajte kod pc rel zapisa.
- void **addToRelTable** (string arg, int **adressing**)
Dodaje relokacioni zapis u tabelu.
- void **fillRelTable** ()
Popunjava relokacionu tabelu.
- void **addList** ()
Dodaje novu listu za relokacione zapise.

Variables

- list< **pomNode** > **pomPCList**
- vector< list< **relNode** > > **relTables**
Relokacioni zapisi.
- int **nextLC** = 0

4.11.1 Function Documentation

4.11.1.1 addList()

```
void addList ( )
```

Dodaje novu listu za relokacione zapise.

4.11.1.2 addToPomPcList()

```
void addToPomPcList (
    int lc,
    int next,
    string symbol,
    int section )
```

Dodaje u listu koja pomaze u racunanju pomeraja za pc rel.

4.11.1.3 addToRelTable()

```
void addToRelTable (
    string arg,
    int adressing )
```

Dodaje relokacioni zapis u tabelu.

4.11.1.4 fillPomBytes()

```
void fillPomBytes ( )
```

Popunjava bajte kod pc rel zapisa.

4.11.1.5 fillRelTable()

```
void fillRelTable ( )
```

Popunjava relokacionu tabelu.

4.11.2 Variable Documentation

4.11.2.1 nextLC

```
int nextLC = 0
```

4.11.2.2 pomPCList

```
list< pomNode> pomPCList
```

4.11.2.3 relTables

```
vector< list< relNode> > relTables
```

Relokacioni zapisi.

4.12 C:/Users/DUSAN/Desktop/ss PROJEKAT/Relocation.h File Reference

```
#include <string>
#include <list>
#include <vector>
```

Classes

- struct **pomNode**
Cvor u listi koja služi da racuna .equ izraze.
- struct **relNode**
Cvor u listi koja čuva relokacione zapise.

Functions

- void **addToRelTable** (string arg, int **adressing**)
Dodaje relokacioni zapis u tabelu.
- void **addList** ()
Dodaje novu listu za relokacione zapise.
- void **addToPomPcList** (int lc, int next, string symbol, int **section**)
Dodaje u listu koja pomaze u racunanju pomeraja za pc rel.
- void **fillPomBytes** ()
Popunjava bajte kod pc rel zapisa.
- void **fillRelTable** ()
Popunjava relokacionu tabelu.

Variables

- vector< list< **relNode** > > **relTables**
Relokacioni zapisi.
- list< **pomNode** > **pomPCList**
- int **nextLC**

4.12.1 Function Documentation

4.12.1.1 `addList()`

```
void addList ( )
```

Dodaje novu listu za relokacione zapise.

4.12.1.2 `addToPomPcList()`

```
void addToPomPcList (
    int lc,
    int next,
    string symbol,
    int section )
```

Dodaje u listu koja pomaze u racunanju pomeraja za pc rel.

4.12.1.3 `addToRelTable()`

```
void addToRelTable (
    string arg,
    int adressing )
```

Dodaje relokacioni zapis u tabelu.

4.12.1.4 `fillPomBytes()`

```
void fillPomBytes ( )
```

Popunjava bajte kod pc rel zapisa.

4.12.1.5 `fillRelTable()`

```
void fillRelTable ( )
```

Popunjava relokacionu tabelu.

4.12.2 Variable Documentation

4.12.2.1 nextLC

```
int nextLC [extern]
```

4.12.2.2 pomPCList

```
list< pomNode> pomPCList [extern]
```

4.12.2.3 relTables

```
vector< list< relNode> > relTables [extern]
```

Relokacioni zapisi.

4.13 C:/Users/DUSAN/Desktop/ss PROJEKAT/SectionTable.cpp File Reference

```
#include "SectionTable.h"  
#include <iostream>
```

Functions

- void **setSection** (string name, int **locationCounter**)
Postavlja sekciju u tabeli sekcija.
- int **findLC** (int **section**)
Trazi location counter na kome se nalazi sekcija.

Variables

- list< **sectionNode** > **sectionTable**
Tabela sekcija.

4.13.1 Function Documentation

4.13.1.1 findLC()

```
int findLC (
    int section )
```

Trazi location counter na kome se nalazi sekcija.

4.13.1.2 setSection()

```
void setSection (
    string name,
    int locationCounter )
```

Postavlja sekciju u tabeli sekcija.

4.13.2 Variable Documentation

4.13.2.1 sectionTable

```
list< sectionNode> sectionTable
```

Tabela sekcija.

4.14 C:/Users/DUSAN/Desktop/ss PROJEKAT/SectionTable.h File Reference

```
#include <iostream>
#include <list>
#include <string>
```

Classes

- struct **sectionNode**
Cvor u listi za sekcije.

Functions

- void **setSection** (string name, int **locationCounter**)
Postavlja sekciju u tabeli sekcija.
- int **findLC** (int **section**)
Trazi location counter na kome se nalazi sekcija.

Variables

- `list< sectionNode > sectionTable`

Tabela sekcija.

4.14.1 Function Documentation

4.14.1.1 findLC()

```
int findLC (
    int section )
```

Trazi location counter na kome se nalazi sekcija.

4.14.1.2 setSection()

```
void setSection (
    string name,
    int locationCounter )
```

Postavlja sekciju u tabeli sekcija.

4.14.2 Variable Documentation

4.14.2.1 sectionTable

```
list< sectionNode> sectionTable [extern]
```

Tabela sekcija.

4.15 C:/Users/DUSAN/Desktop/ss PROJEKAT/SymbolTable.cpp File Reference

```
#include <string>
#include <iostream>
#include <list>
#include <regex>
#include "SymbolTable.h"
#include "FlinkStruct.h"
#include "RegexSearch.h"
#include "DigitConvert.h"
#include "SectionTable.h"
#include "outputData.h"
```

Functions

- void **setSymbol** (string arg, int &lc, char c, int **section**)
Postavlja simbol u tabeli simbola.
- void **setSymbolLabel** (string arg, int &lc, int **section**, bool **equ**, int equs)
Dodaje labelu u tabelu simbola.
- void **setSymbolExtern** (string arg, char c)
Dodaje extern simbol u tabelu simbola.
- void **setSymbolGlobal** (string arg, char c, int **section**)
Cvor u listi koja služi da racuna .equ izraze.
- void **writeList** (list< **Flink_Node** > lista)
Cvor u listi koja služi da racuna .equ izraze.
- bool **findSymbol** (string name, int &symbol)
Traži simbol u tabeli simbola.
- bool **findSymbolSEC** (string name, int &symbol, int &sec)
Traži sekciju datog simbola.
- bool **findSymbolBP** (string name, int &symbol, int &size)
Traži simbol za backpatching.
- void **findSymbolIPC** (string name, int &value, int & **section**, char &c)
Traži simbol za pc rel adresiranje i sredjivanje masinskog koda.
- void **findSymbolREL** (string name, int & **section**, char &LG)
Traži simbol kod racunanja equ izraza.
- void **findSymbolRELOCATION** (string name, int & **section**, char &LG, bool &isEqu, int &equSection)
Traži simbol za u tabelu relokacija.
- void **backpatching** ()
Funkcija koja vrsi backpatching.
- void **changeLC** ()
Menja location counter.

Variables

- list< **symbolNode** > **symbol_Table**
- vector< string > **opCode**
- bool **equ** = false
- int **locationCounter** = 0
- int **section** = 0
- bool **equSearched** = false

4.15.1 Function Documentation

4.15.1.1 backpatching()

```
void backpatching ( )
```

Funkcija koja vrsi backpatching.

4.15.1.2 changeLC()

```
void changeLC ( )
```

Menja location counter.

4.15.1.3 findSymbol()

```
bool findSymbol (
    string name,
    int & symbol )
```

Trazi simbol u tabeli simbola.

4.15.1.4 findSymbolBP()

```
bool findSymbolBP (
    string name,
    int & symbol,
    int & size )
```

Trazi simbol za backpatching.

4.15.1.5 findSymbolPC()

```
void findSymbolPC (
    string name,
    int & value,
    int & section,
    char & c )
```

Trazi simbol za pc rel adresiranje i sredjivanje masinskog koda.

4.15.1.6 findSymbolREL()

```
void findSymbolREL (
    string name,
    int & section,
    char & LG )
```

Trazi simol kod racunanja equ izraza.

4.15.1.7 findSymbolRELOCATION()

```
void findSymbolRELOCATION (
    string name,
    int & section,
    char & LG,
    bool & isEqu,
    int & equSection )
```

Trazi simbol za u tabelu relokacija.

4.15.1.8 findSymbolSEC()

```
bool findSymbolSEC (
    string name,
    int & symbol,
    int & sec )
```

Trazi sekciju datog simbola.

4.15.1.9 setSymbol()

```
void setSymbol (
    string arg,
    int & lc,
    char c,
    int section )
```

Postavlja simbol u tabeli simbola.

4.15.1.10 setSymbolExtern()

```
void setSymbolExtern (
    string arg,
    char c )
```

Dodaje extern simbol u tabelu simbola.

4.15.1.11 setSymbolGlobal()

```
void setSymbolGlobal (
    string arg,
    char c,
    int section )
```

Cvor u listi koja služi da računa .equ izraze.

4.15.1.12 setSymbolLabel()

```
void setSymbolLabel (
    string arg,
    int & lc,
    int section,
    bool equ,
    int equs )
```

Dodaje labelu u tabelu simbola.

4.15.1.13 writeList()

```
void writeList (
    list< Flink_Node > lista )
```

Cvor u listi koja služi da racuna .equ izraze.

4.15.2 Variable Documentation

4.15.2.1 equ

```
bool equ = false
```

4.15.2.2 equSearched

```
bool equSearched = false
```

4.15.2.3 locationCounter

```
int locationCounter = 0
```

4.15.2.4 opCode

```
vector<string> opCode
```

4.15.2.5 section

```
int section = 0
```

4.15.2.6 symbol_Table

```
list< symbolNode> symbol_Table
```

4.16 C:/Users/DUSAN/Desktop/ss PROJEKAT/SymbolTable.h File Reference

```
#include <string>
#include <list>
#include <vector>
#include "FlinkStruct.h"
```

Classes

- struct **symbolNode**
Covr u listi rabele simbola.

Functions

- void **setSymbol** (string arg, int &lc, char c, int **section**)
Postavlja simbol u tabeli simbola.
- void **setSymbolLabel** (string arg, int &lc, int **section**, bool **equ**, int equsection)
Dodaje labelu u tabelu simbola.
- void **setSymbolExtern** (string arg, char c)
Dodaje extern simbol u tabelu simbola.
- void **changeLC** ()
Menja location counter.
- void **setSymbolGlobal** (string arg, char c, int **section**)
Cvor u listi koja služi da racuna .equ izraze.
- void **writeList** (list< **Flink_Node** > lista)
Cvor u listi koja služi da racuna .equ izraze.
- bool **findSymbol** (string name, int &symbol)
Trazi simbol u tabeli simbola.
- bool **findSymbolSEC** (string name, int &symbol, int &sec)
Trazi sekciju datog simbola.
- bool **findSymbolBP** (string name, int &symbol, int &size)
Trazi simbol za backpatching.
- void **findSymbolPC** (string name, int &value, int &**section**, char &c)
Trazi simbol za pc rel adresiranje i sredjivanje masinskog koda.
- void **findSymbolREL** (string name, int &**section**, char &LG)
Trazi simbol kod racunanja equ izraza.
- void **findSymbolRELOCATION** (string name, int &**section**, char &LG, bool &isEqu, int &equSection)
Trazi simbol za u tabelu relokacija.
- void **backpatching** ()
Funkcija koja vrsi backpatching.

Variables

- `list< symbolNode > symbol_Table`
- `vector< string > opCode`
- `int locationCounter`
- `bool equSearched`
- `int section`
- `bool equ`

4.16.1 Function Documentation

4.16.1.1 backpatching()

```
void backpatching ( )
```

Funkcija koja vrši backpatching.

4.16.1.2 changeLC()

```
void changeLC ( )
```

Menja location counter.

4.16.1.3 findSymbol()

```
bool findSymbol (
    string name,
    int & symbol )
```

Traži simbol u tabeli simbola.

4.16.1.4 findSymbolBP()

```
bool findSymbolBP (
    string name,
    int & symbol,
    int & size )
```

Traži simbol za backpatching.

4.16.1.5 findSymbolPC()

```
void findSymbolPC (
    string name,
    int & value,
    int & section,
    char & c )
```

Trazi simbol za pc rel adresiranje i sredjivanje masinskog koda.

4.16.1.6 findSymbolREL()

```
void findSymbolREL (
    string name,
    int & section,
    char & LG )
```

Trazi simol kod racunanja equ izraza.

4.16.1.7 findSymbolRELOCATION()

```
void findSymbolRELOCATION (
    string name,
    int & section,
    char & LG,
    bool & isEqu,
    int & equSection )
```

Trazi simbol za u tabelu relokacija.

4.16.1.8 findSymbolSEC()

```
bool findSymbolSEC (
    string name,
    int & symbol,
    int & sec )
```

Trazi sekciju datog simbola.

4.16.1.9 setSymbol()

```
void setSymbol (
    string arg,
    int & lc,
    char c,
    int section )
```

Postavlja simbol u tabeli simbola.

4.16.1.10 setSymbolExtern()

```
void setSymbolExtern (
    string arg,
    char c )
```

Dodaje extern simbol u tabelu simbola.

4.16.1.11 setSymbolGlobal()

```
void setSymbolGlobal (
    string arg,
    char c,
    int section )
```

Cvor u listi koja služi da računa .equ izraze.

4.16.1.12 setSymbolLabel()

```
void setSymbolLabel (
    string arg,
    int & lc,
    int section,
    bool equ,
    int equsection )
```

Dodaje labelu u tabelu simbola.

4.16.1.13 writeList()

```
void writeList (
    list< Flink_Node > lista )
```

Cvor u listi koja služi da računa .equ izraze.

4.16.2 Variable Documentation

4.16.2.1 equ

```
bool equ [extern]
```

4.16.2.2 equSearched

```
bool equSearched [extern]
```

4.16.2.3 locationCounter

```
int locationCounter [extern]
```

4.16.2.4 opCode

```
vector<string> opCode [extern]
```

4.16.2.5 section

```
int section [extern]
```

4.16.2.6 symbol_Table

```
list< symbolNode> symbol_Table [extern]
```


Index

- addList
 - Relocation.cpp, 33
 - Relocation.h, 36
- addToPomPcList
 - Relocation.cpp, 33
 - Relocation.h, 36
- addToRelTable
 - Relocation.cpp, 34
 - Relocation.h, 36
- adress
 - Flink_Node, 6
- addressing
 - RegexSearch.cpp, 23
 - RegexSearch.h, 28
 - relNode, 8
- Adressing.cpp
 - determinateAddressing, 14
 - immed, 14
 - memdir, 14
 - regdir, 14
 - regind, 14
 - regindpom, 14
 - sym, 15
- Adressing.h
 - determinateAddressing, 15
 - immed, 16
 - memdir, 16
 - regdir, 16
 - regind, 16
 - regindpom, 16
 - sym, 17
- backpatching
 - SymbolTable.cpp, 40
 - SymbolTable.h, 45
- C:/Users/DUSAN/Desktop/ss PROJEKAT/Adressing.cpp, 13
- C:/Users/DUSAN/Desktop/ss PROJEKAT/Adressing.h, 15
- C:/Users/DUSAN/Desktop/ss PROJEKAT/DigitConvert.cpp, 17
- C:/Users/DUSAN/Desktop/ss PROJEKAT/DigitConvert.h, 18
- C:/Users/DUSAN/Desktop/ss PROJEKAT/FlinkStruct.h, 19
- C:/Users/DUSAN/Desktop/ss PROJEKAT/main.cpp, 19
- C:/Users/DUSAN/Desktop/ss PROJEKAT/outputData.cpp, 20
- C:/Users/DUSAN/Desktop/ss PROJEKAT/outputData.h, 21
- C:/Users/DUSAN/Desktop/ss PROJEKAT/RegexSearch.cpp, 22
- C:/Users/DUSAN/Desktop/ss PROJEKAT/RegexSearch.h, 27
- C:/Users/DUSAN/Desktop/ss PROJEKAT/Relocation.cpp, 33
- C:/Users/DUSAN/Desktop/ss PROJEKAT/Relocation.h, 35
- C:/Users/DUSAN/Desktop/ss PROJEKAT/SectionTable.cpp, 37
- C:/Users/DUSAN/Desktop/ss PROJEKAT/SectionTable.h, 38
- C:/Users/DUSAN/Desktop/ss PROJEKAT/SymbolTable.cpp, 39
- C:/Users/DUSAN/Desktop/ss PROJEKAT/SymbolTable.h, 44
- changeArg
 - RegexSearch.cpp, 23
 - RegexSearch.h, 29
- changeLC
 - SymbolTable.cpp, 40
 - SymbolTable.h, 45
- countAllEques
 - RegexSearch.cpp, 24
 - RegexSearch.h, 29
- countEqu
 - RegexSearch.cpp, 24
 - RegexSearch.h, 29
- defined
 - symbolNode, 11
- determinateAddressing
 - Adressing.cpp, 14
 - Adressing.h, 15
- DigitConvert.cpp
 - intToHex, 17
 - makeTwoBytes, 17
 - makeWord, 18
- DigitConvert.h
 - intToHex, 18
 - makeTwoBytes, 18
 - makeWord, 19
- equ
 - SymbolTable.cpp, 43
 - SymbolTable.h, 48
- equation
 - equNode, 5

- equList
 - RegexSearch.cpp, 26
 - RegexSearch.h, 32
- equNode, 5
 - equation, 5
 - s, 5
 - section, 5
 - symbol, 6
- equSearched
 - SymbolTable.cpp, 43
 - SymbolTable.h, 48
- equSection
 - symbolNode, 11
- fillPomBytes
 - Relocation.cpp, 34
 - Relocation.h, 36
- fillRelTable
 - Relocation.cpp, 34
 - Relocation.h, 36
- findLC
 - SectionTable.cpp, 37
 - SectionTable.h, 39
- findSymbol
 - SymbolTable.cpp, 41
 - SymbolTable.h, 45
- findSymbolBP
 - SymbolTable.cpp, 41
 - SymbolTable.h, 45
- findSymbolIPC
 - SymbolTable.cpp, 41
 - SymbolTable.h, 45
- findSymbolIREL
 - SymbolTable.cpp, 41
 - SymbolTable.h, 46
- findSymbolIRELOCATION
 - SymbolTable.cpp, 41
 - SymbolTable.h, 46
- findSymbolSEC
 - SymbolTable.cpp, 42
 - SymbolTable.h, 46
- firstArg
 - RegexSearch.cpp, 26
 - RegexSearch.h, 32
- Flink
 - symbolNode, 11
- Flink_Node, 6
 - adress, 6
 - op, 6
- GL
 - sectionNode, 10
- immed
 - Addressing.cpp, 14
 - Addressing.h, 16
- inImmed
 - RegexSearch.cpp, 27
 - RegexSearch.h, 32
- intToHex
 - DigitConvert.cpp, 17
 - DigitConvert.h, 18
- isEqu
 - symbolNode, 12
- jump
 - RegexSearch.cpp, 27
 - RegexSearch.h, 32
- LG
 - symbolNode, 12
- locationCounter
 - SymbolTable.cpp, 43
 - SymbolTable.h, 48
- locationCounter_
 - pomNode, 7
- lowHigh
 - RegexSearch.cpp, 27
 - RegexSearch.h, 32
- main
 - main.cpp, 20
- main.cpp
 - main, 20
- makeFirstByte
 - RegexSearch.cpp, 24
 - RegexSearch.h, 29
- makeSecondByte
 - RegexSearch.cpp, 24
 - RegexSearch.h, 29
- makeTwoBytes
 - DigitConvert.cpp, 17
 - DigitConvert.h, 18
- makeWord
 - DigitConvert.cpp, 18
 - DigitConvert.h, 19
- memdir
 - Addressing.cpp, 14
 - Addressing.h, 16
- name
 - sectionNode, 10
 - symbolNode, 12
- nextLC
 - pomNode, 7
 - Relocation.cpp, 34
 - Relocation.h, 37
- offset
 - relNode, 9
 - sectionNode, 10
- oneArg
 - RegexSearch.cpp, 27
 - RegexSearch.h, 32
- op
 - Flink_Node, 6
- opCode
 - SymbolTable.cpp, 43

- SymbolTable.h, 48
- outputData.cpp
 - writeData, 20
 - writeErrorDest, 20
 - writeErrorNotDefined, 21
 - writeExternError, 21
- outputData.h
 - writeData, 21
 - writeErrorDest, 21
 - writeErrorNotDefined, 22
 - writeExternError, 22
- pomNode, 7
 - locationCounter_, 7
 - nextLC, 7
 - pomNode, 7
 - section_, 8
 - symbol_, 8
- pomPCList
 - Relocation.cpp, 34
 - Relocation.h, 37
- putInEquList
 - RegexSearch.cpp, 24
 - RegexSearch.h, 30
- realLC
 - sectionNode, 10
- regdir
 - Addressing.cpp, 14
 - Addressing.h, 16
- RegexSearch.cpp
 - adressing, 23
 - changeArg, 23
 - countAllEques, 24
 - countEqu, 24
 - equList, 26
 - firstArg, 26
 - inImmed, 27
 - jump, 27
 - lowHigh, 27
 - makeFirstByte, 24
 - makeSecondByte, 24
 - oneArg, 27
 - putInEquList, 24
 - searchEG, 25
 - searchEqu, 25
 - searchIns, 25
 - searchIns2, 25
 - searchInsNoArg, 25
 - searchLabel, 26
 - searchSection, 26
 - searchSWB, 26
- RegexSearch.h
 - adressing, 28
 - changeArg, 29
 - countAllEques, 29
 - countEqu, 29
 - equList, 32
 - firstArg, 32
 - inImmed, 32
 - jump, 32
 - lowHigh, 32
 - makeFirstByte, 29
 - makeSecondByte, 29
 - oneArg, 32
 - putInEquList, 30
 - searchEG, 30
 - searchEqu, 30
 - searchIns, 30
 - searchIns2, 30
 - searchInsNoArg, 31
 - searchLabel, 31
 - searchSection, 31
 - searchSWB, 31
- regind
 - Addressing.cpp, 14
 - Addressing.h, 16
- regindpom
 - Addressing.cpp, 14
 - Addressing.h, 16
- relNode, 8
 - adressing, 8
 - offset, 9
 - section, 9
 - symbol, 9
 - type, 9
- Relocation.cpp
 - addList, 33
 - addToPomPcList, 33
 - addToRelTable, 34
 - fillPomBytes, 34
 - fillRelTable, 34
 - nextLC, 34
 - pomPCList, 34
 - relTables, 35
- Relocation.h
 - addList, 36
 - addToPomPcList, 36
 - addToRelTable, 36
 - fillPomBytes, 36
 - fillRelTable, 36
 - nextLC, 37
 - pomPCList, 37
 - relTables, 37
- relTables
 - Relocation.cpp, 35
 - Relocation.h, 37
- rwa
 - sectionNode, 10
- s
 - equNode, 5
- searchEG
 - RegexSearch.cpp, 25
 - RegexSearch.h, 30
- searchEqu
 - RegexSearch.cpp, 25
 - RegexSearch.h, 30

- searchIns
 - RegexSearch.cpp, 25
 - RegexSearch.h, 30
- searchIns2
 - RegexSearch.cpp, 25
 - RegexSearch.h, 30
- searchInsNoArg
 - RegexSearch.cpp, 25
 - RegexSearch.h, 31
- searchLabel
 - RegexSearch.cpp, 26
 - RegexSearch.h, 31
- searchSection
 - RegexSearch.cpp, 26
 - RegexSearch.h, 31
- searchSWB
 - RegexSearch.cpp, 26
 - RegexSearch.h, 31
- section
 - equNode, 5
 - relNode, 9
 - symbolNode, 12
 - SymbolTable.cpp, 43
 - SymbolTable.h, 48
- section_
 - pomNode, 8
- sectionNode, 9
 - GL, 10
 - name, 10
 - offset, 10
 - realLC, 10
 - rwa, 10
- sectionTable
 - SectionTable.cpp, 38
 - SectionTable.h, 39
- SectionTable.cpp
 - findLC, 37
 - sectionTable, 38
 - setSection, 38
- SectionTable.h
 - findLC, 39
 - sectionTable, 39
 - setSection, 39
- setSection
 - SectionTable.cpp, 38
 - SectionTable.h, 39
- setSymbol
 - SymbolTable.cpp, 42
 - SymbolTable.h, 46
- setSymbolExtern
 - SymbolTable.cpp, 42
 - SymbolTable.h, 47
- setSymbolGlobal
 - SymbolTable.cpp, 42
 - SymbolTable.h, 47
- setSymbolLabel
 - SymbolTable.cpp, 42
 - SymbolTable.h, 47
- size
 - symbolNode, 12
- sym
 - Addressing.cpp, 15
 - Addressing.h, 17
- symbol
 - equNode, 6
 - relNode, 9
- symbol_
 - pomNode, 8
- symbol_Table
 - SymbolTable.cpp, 44
 - SymbolTable.h, 48
- symbolNode, 10
 - defined, 11
 - equSection, 11
 - Flink, 11
 - isEqu, 12
 - LG, 12
 - name, 12
 - section, 12
 - size, 12
 - symbolNode, 11
 - value, 12
- SymbolTable.cpp
 - backpatching, 40
 - changeLC, 40
 - equ, 43
 - equSearched, 43
 - findSymbol, 41
 - findSymbolBP, 41
 - findSymbolPC, 41
 - findSymbolREL, 41
 - findSymbolRELOCATION, 41
 - findSymbolSEC, 42
 - locationCounter, 43
 - opCode, 43
 - section, 43
 - setSymbol, 42
 - setSymbolExtern, 42
 - setSymbolGlobal, 42
 - setSymbolLabel, 42
 - symbol_Table, 44
 - writeList, 43
- SymbolTable.h
 - backpatching, 45
 - changeLC, 45
 - equ, 48
 - equSearched, 48
 - findSymbol, 45
 - findSymbolBP, 45
 - findSymbolPC, 45
 - findSymbolREL, 46
 - findSymbolRELOCATION, 46
 - findSymbolSEC, 46
 - locationCounter, 48
 - opCode, 48
 - section, 48

- setSymbol, 46
- setSymbolExtern, 47
- setSymbolGlobal, 47
- setSymbolLabel, 47
- symbol_Table, 48
- writeList, 47

type

- relNode, 9

value

- symbolNode, 12

writeData

- outputData.cpp, 20
- outputData.h, 21

writeErrorDest

- outputData.cpp, 20
- outputData.h, 21

writeErrorNotDefined

- outputData.cpp, 21
- outputData.h, 22

writeExternError

- outputData.cpp, 21
- outputData.h, 22

writeList

- SymbolTable.cpp, 43
- SymbolTable.h, 47