MIPS 32 simulator From Scratch

0.0.1

Generated by Doxygen 1.7.6.1

Mon Nov 4 2013 17:49:30

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 Elf_Arhdr Struct Reference

```
#include <libelf.h>
```

Data Fields

- char ∗ ar_name
- time_t ar_date
- uid_t ar_uid
- gid_t ar_gid
- mode_t ar_mode
- loff_t ar_size
- char ∗ ar_rawname

3.1.1 Detailed Description

Definition at line 153 of file libelf.h.

3.1.2 Field Documentation

3.1.2.1 time_t ar_date

Definition at line 156 of file libelf.h.

3.1.2.2 gid_t ar_gid

Definition at line 158 of file libelf.h.

3.1.2.3 mode_t ar_mode

Definition at line 159 of file libelf.h.

3.1.2.4 char∗ ar_name

Definition at line 155 of file libelf.h.

3.1.2.5 char∗ ar_rawname

Definition at line 161 of file libelf.h.

3.1.2.6 loff_t ar_size

Definition at line 160 of file libelf.h.

3.1.2.7 uid_t ar_uid

Definition at line 157 of file libelf.h.

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

- /home/ben_jah/Bureau/Projet_informatique/step2/include/libelf.h

3.2 Elf_Arsym Struct Reference

```
#include <libelf.h>
```

Data Fields

- char ∗ as_name
- size_t as_off
- unsigned long int as_hash

3.2.1 Detailed Description

Definition at line 166 of file libelf.h.

3.2.2 Field Documentation

3.2.2.1 unsigned long int as_hash

Definition at line 170 of file libelf.h.

3.2.2.2 char∗ as_name

Definition at line 168 of file libelf.h.

3.2.2.3 size_t as_off

Definition at line 169 of file libelf.h.

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

- /home/ben_jah/Bureau/Projet_informatique/step2/include/libelf.h

3.3 Elf_Data Struct Reference

```
#include <libelf.h>
```

Data Fields

- void ∗ d_buf
- Elf_Type d_type
- unsigned int d_version
- size_t d_size
- loff_t d_off
- size_t d_align

3.3.1 Detailed Description

Definition at line 92 of file libelf.h.

3.3.2 Field Documentation

3.3.2.1 size_t d_align

Definition at line 99 of file libelf.h.

3.3.2.2 void∗ d_buf

Definition at line 94 of file libelf.h.

3.3.2.3 loff_t d_off

Definition at line 98 of file libelf.h.

3.3.2.4 size_t d_size

Definition at line 97 of file libelf.h.

3.3.2.5 Elf_Type d_type

Definition at line 95 of file libelf.h.

3.3.2.6 unsigned int d_version

Definition at line 96 of file libelf.h.

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

- /home/ben_jah/Bureau/Projet_informatique/step2/include/libelf.h

3.4 MemZone Struct Reference

Type de donn zone ! Definition des informations d'une section ELF : (nom, donn, taille des donn, adresse mire de la section, ...) + acces aux differentes sections du fichier ELF. Cette structure de donn est completement dndante de la librairie ELF.

```
#include <mipself.h>
```

Data Fields

- Elf_Scn ∗ scn
- char ∗ name
- Elf32_Half index
- Elf32_Half type
- Elf32_Word size
- Elf32_Word mem_start
- Elf_Scn ∗ rel_scn
- char ∗ rel_name
- unsigned char ∗ data
- SectionELF ∗ exportSection

3.4.1 Detailed Description

Type de donn zone ! Definition des informations d'une section ELF : (nom, donn, taille des donn, adresse mire de la section, ...) + acces aux differentes sections du fichier ELF. Cette structure de donn est completement dndante de la librairie ELF.

Definition at line 49 of file mipself.h.

3.4.2 Field Documentation

3.4.2.1 unsigned char* data

Definition at line 61 of file mipself.h.

3.4.2.2 SectionELF* exportSection

Definition at line 63 of file mipself.h.

3.4.2.3 Elf32_Half index

Definition at line 54 of file mipself.h.

3.4.2.4 Elf32_Word mem_start

Definition at line 57 of file mipself.h.

3.4.2.5 char* name

Definition at line 53 of file mipself.h.

3.4.2.6 char* rel_name

Definition at line 60 of file mipself.h.

3.4.2.7 Elf_Scn* rel_scn

Definition at line 58 of file mipself.h.

3.4.2.8 Elf_Scn* scn

Definition at line 51 of file mipself.h.

3.4.2.9 Elf32_Word size

Definition at line 56 of file mipself.h.

3.4.2.10 Elf32_Half type

Definition at line 55 of file mipself.h.

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

- /home/ben_jah/Bureau/Projet_informatique/step2/include/mipself.h

3.5 mips Struct Reference

```
#include <structure.h>
```

Data Fields

- registre reg [35]
- SectionELF segment [3]

3.5.1 Detailed Description

Definition at line 25 of file structure.h.

3.5.2 Field Documentation

3.5.2.1 registre reg[35]

Definition at line 26 of file structure.h.

3.5.2.2 SectionELF segment[3]

Definition at line 27 of file structure.h.

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

- /home/ben_jah/Bureau/Projet_informatique/step2/include/structure.h

3.6 nodeSymbol Struct Reference

Data Fields

- unsigned long adr
- char ∗ name
- struct nodeSymbol ∗ suiv

3.6.1 Detailed Description

Definition at line 265 of file mipself.c.

3.6.2 Field Documentation

3.6.2.1 unsigned long adr

Definition at line 266 of file mipself.c.

3.6.2.2 char∗ name

Definition at line 267 of file mipself.c.

3.6.2.3 struct nodeSymbol∗ suiv

Definition at line 268 of file mipself.c.

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

- /home/ben_jah/Bureau/Projet_informatique/step2/src/mipself.c

3.7 registre Struct Reference

```
#include <structure.h>
```

Data Fields

- unsigned int indice
- char ∗ mnemo
- unsigned int val

3.7.1 Detailed Description

Definition at line 19 of file structure.h.

3.7.2 Field Documentation

3.7.2.1 unsigned int indice

Definition at line 20 of file structure.h.

3.7.2.2 char∗ mnemo

Definition at line 21 of file structure.h.

3.7.2.3 unsigned int val

Definition at line 22 of file structure.h.

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

- /home/ben_jah/Bureau/Projet_informatique/step2/include/structure.h

3.8 SectionELF Struct Reference

Type de donn SectionELF ! Definition des informations d'une section ELF : (nom, donn, taille des donn, adresse mire de la section, ...) Cette structure d'ange, quasi-ivalente structure MemZone dnie dans mipsloader.c, est indndante de la librairie ELF. Elle est est utilisour les interfaces avec le shell!

```
#include <mipself.h>
```

Data Fields

- char ∗ name
- WORD size
- BYTE ∗ data
- WORD startAddress

3.8.1 Detailed Description

Type de donn SectionELF ! Definition des informations d'une section ELF : (nom, donn, taille des donn, adresse mire de la section, ...) Cette structure d'ange, quasi-ivalente structure MemZone dnie dans mipsloader.c, est indndante de la librairie ELF. Elle est est utilisour les interfaces avec le shell!

Definition at line 32 of file mipself.h.

3.8.2 Field Documentation

3.8.2.1 BYTE∗ data

Definition at line 35 of file mipself.h.

3.8.2.2 char∗ name

Definition at line 33 of file mipself.h.

3.8.2.3 WORD size

Definition at line 34 of file mipself.h.

3.8.2.4 WORD startAddress

Definition at line 36 of file mipself.h.

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

- /home/ben_jah/Bureau/Projet_informatique/step2/include/mipself.h

Chapter 4

File Documentation

4.1 /home/ben_jah/Bureau/Projet_informatique/step2/include/cmdSearch.h File Reference

Functions

- char ∗ cmdSearch (char ∗type, unsigned int param)

4.1.1 Function Documentation

4.1.1.1 char∗ cmdSearch (char ∗ *type*, unsigned int *param*)

Definition at line 27 of file cmdSearch.c.

4.2 /home/ben_jah/Bureau/Projet_informatique/step2/include/constantes.h File Reference

Defines

- #define MAX_STR 1024
- #define CMD_OK_RETURN_VALUE 0
- #define CMD_EXIT_RETURN_VALUE -1
- #define CMD_EMPTY_RETURN_VALUE -2
- #define CMD_UNKOWN_RETURN_VALUE -3
- #define PROMPT_STRING "SimMipsShell : > "

4.2.1 Define Documentation

4.2.1.1 #define CMD_EMPTY_RETURN_VALUE -2

Definition at line 11 of file constantes.h.

4.2.1.2 #define CMD_EXIT_RETURN_VALUE -1

Definition at line 10 of file constantes.h.

4.2.1.3 #define CMD_OK_RETURN_VALUE 0

Definition at line 9 of file constantes.h.

4.2.1.4 #define CMD_UNKOWN_RETURN_VALUE -3

Definition at line 12 of file constantes.h.

4.2.1.5 #define MAX_STR 1024

Definition at line 6 of file constantes.h.

4.2.1.6 #define PROMPT_STRING "SimMipsShell : > "

Definition at line 19 of file constantes.h.

4.3 /home/ben_jah/Bureau/Projet_informatique/step2/include/elfimport.h File Reference

```
#include <libelf.h>
```

Defines

- #define R_MIPS_NONE 0

  Importation de la librairie LIBELF.
- #define R_MIPS_16 1
- #define R_MIPS_32 2
- #define R_MIPS_REL32 3
- #define R_MIPS_26 4
- #define R_MIPS_HI16 5
- #define R_MIPS_LO16 6
- #define R_MIPS_GPREL16 7
- #define R_MIPS_LITERAL 8
- #define R_MIPS_GOT16 9

- #define R_MIPS_PC16 10
- #define R_MIPS_CALL16 11
- #define R_MIPS_GPREL32 12
- #define R_MIPS_GOTHI16 21
- #define R_MIPS_GOTLO16 22
- #define R_MIPS_CALLHI16 30
- #define R_MIPS_CALLLO16 31

4.3.1 Define Documentation

4.3.1.1 #define R_MIPS_16 1

Definition at line 15 of file elfimport.h.

4.3.1.2 #define R_MIPS_26 4

Definition at line 18 of file elfimport.h.

4.3.1.3 #define R_MIPS_32 2

Definition at line 16 of file elfimport.h.

4.3.1.4 #define R_MIPS_CALL16 11

Definition at line 25 of file elfimport.h.

4.3.1.5 #define R_MIPS_CALLHI16 30

Definition at line 29 of file elfimport.h.

4.3.1.6 #define R_MIPS_CALLLO16 31

Definition at line 30 of file elfimport.h.

4.3.1.7 #define R_MIPS_GOT16 9

Definition at line 23 of file elfimport.h.

4.3.1.8 #define R_MIPS_GOTHI16 21

Definition at line 27 of file elfimport.h.

4.3.1.9 #define R_MIPS_GOTLO16 22

Definition at line 28 of file elfimport.h.

4.3.1.10 #define R_MIPS_GPREL16 7

Definition at line 21 of file elfimport.h.

4.3.1.11 #define R_MIPS_GPREL32 12

Definition at line 26 of file elfimport.h.

4.3.1.12 #define R_MIPS_HI16 5

Definition at line 19 of file elfimport.h.

4.3.1.13 #define R_MIPS_LITERAL 8

Definition at line 22 of file elfimport.h.

4.3.1.14 #define R_MIPS_LO16 6

Definition at line 20 of file elfimport.h.

4.3.1.15 #define R_MIPS_NONE 0

Importation de la librairie LIBELF.

PHELMA - Projet Informatique

Author

(c) Laurent Fesquet 2003,
(m) Matthieu Chabanas 2005
(m) François Portet 2013 <francois.portet@imag.fr>

Definition at line 14 of file elfimport.h.

4.3.1.16 #define R_MIPS_PC16 10

Definition at line 24 of file elfimport.h.

4.3.1.17 #define R_MIPS_REL32 3

Definition at line 17 of file elfimport.h.

4.4 /home/ben_jah/Bureau/Projet_informatique/step2/include/fonctions.h File Reference

Functions

- int execute_cmd_testcmd (int hexValue)
- int parse_and_execute_cmd_testcmd (char ∗paramsStr, mips ∗arch)
- int execute_cmd_exit ()
- int parse_and_execute_cmd_exit (char ∗paramsStr)
- int parse_and_execute_cmd_string (char ∗input, mips ∗arch)
- int acquire_line (FILE ∗fp, char ∗input)
- void usage_ERROR_MSG (char ∗command)

4.4.1 Function Documentation

4.4.1.1 int **acquire_line** (FILE ∗ *fp,* char ∗ *input*)

Definition at line 146 of file fonctions.c.

4.4.1.2 int **execute_cmd_exit** ( )

Definition at line 59 of file fonctions.c.

4.4.1.3 int **execute_cmd_testcmd** (int *hexValue*)

Definition at line 30 of file fonctions.c.

4.4.1.4 int **parse_and_execute_cmd_exit** (char ∗ *paramsStr*)

Definition at line 66 of file fonctions.c.

4.4.1.5 int **parse_and_execute_cmd_string** (char ∗ *input,* **mips** ∗ *arch*)

Definition at line 79 of file fonctions.c.

4.4.1.6 int **parse_and_execute_cmd_testcmd** (char ∗ *paramsStr,* **mips** ∗ *arch*)

Definition at line 43 of file fonctions.c.

4.4.1.7 void **usage_ERROR_MSG** (char ∗ *command*)

Definition at line 178 of file fonctions.c.

4.5 /home/ben_jah/Bureau/Projet_informatique/step2/include/fonctionsStep1.h File Reference

Functions

- void init_mips (mips ∗arch)
- void init_segment (SectionELF ∗seg, int type)
- int parse_and_execute_cmd_lp (char ∗input, mips ∗arch)
- int execute_cmd_lp (FILE ∗program, char ∗nom_fichier, mips ∗arch)
- int parse_and_execute_cmd_lm (char ∗input, mips ∗arch)
- int execute_cmd_lm (unsigned int adr, unsigned int val, mips ∗arch)
- int parse_and_execute_cmd_dm (char ∗input, mips ∗arch)
- int execute_cmd_dm (int cas, unsigned int addr, unsigned int arg2, mips ∗arch)
- int parse_and_execute_cmd_da (char ∗input, mips ∗arch)
- int execute_cmd_da (unsigned int addr, unsigned int nbr_instr, mips ∗arch)

4.5.1 Function Documentation

4.5.1.1 int execute_cmd_da (unsigned int *addr,* unsigned int *nbr_instr,* mips ∗ *arch*)

Definition at line 141 of file fonctionsStep2.c.

4.5.1.2 int execute_cmd_dm (int *cas,* unsigned int *addr,* unsigned int *arg2,* mips ∗ *arch*)

Definition at line 257 of file fonctionsStep1.c.

4.5.1.3 int execute_cmd_lm (unsigned int *adr,* unsigned int *val,* mips ∗ *arch*)

Definition at line 135 of file fonctionsStep1.c.

4.5.1.4 int execute_cmd_lp (FILE ∗ *program,* char ∗ *nom_fichier,* mips ∗ *arch*)

Definition at line 339 of file fonctionsStep2.c.

4.5.1.5 void init_mips (mips ∗ *arch*)

Definition at line 39 of file fonctionsStep1.c.

4.5.1.6 void init_segment (SectionELF ∗ *seg,* int *type*)

Definition at line 87 of file fonctionsStep1.c.

4.5.1.7 int parse_and_execute_cmd_da (char ∗ *input,* mips ∗ *arch*)

Definition at line 55 of file fonctionsStep2.c.

4.5.1.8 int parse_and_execute_cmd_dm (char ∗ *input,* mips ∗ *arch*)

Definition at line 381 of file fonctionsStep1.c.

4.5.1.9 int parse_and_execute_cmd_lm (char ∗ *input,* mips ∗ *arch*)

Definition at line 164 of file fonctionsStep1.c.

4.5.1.10 int parse_and_execute_cmd_lp (char ∗ *input,* mips ∗ *arch*)

Definition at line 291 of file fonctionsStep2.c.

4.6 /home/ben_jah/Bureau/Projet_informatique/step2/include/fonctionsStep2.h File Reference

Functions

- unsigned int getbits (unsigned int val, unsigned int start, unsigned int stop)
- int parse_and_execute_cmd_lr (char ∗input, mips ∗arch)
- int execute_cmd_lr (int adr, unsigned int val, mips ∗arch)
- int parse_and_execute_cmd_dr (char ∗input, mips ∗arch)
- int execute_cmd_dr (int test_reg, mips ∗arch)
- char ∗ cmdSearch (char ∗type, unsigned int param)

4.6.1 Function Documentation

4.6.1.1 char∗ cmdSearch (char ∗ *type,* unsigned int *param*)

Definition at line 27 of file cmdSearch.c.

4.6.1.2 int execute_cmd_dr (int *test_reg,* mips ∗ *arch*)

Definition at line 488 of file fonctionsStep1.c.

4.6.1.3 int execute_cmd_lr (int *adr,* unsigned int *val,* mips ∗ *arch*)

Definition at line 550 of file fonctionsStep1.c.

4.6.1.4 unsigned int **getbits** (unsigned int *val*, unsigned int *start*, unsigned int *stop*)

Definition at line 26 of file fonctionsStep2.c.

4.6.1.5 int **parse_and_execute_cmd_dr** (char ∗ *input*, **mips** ∗ *arch*)

Definition at line 519 of file fonctionsStep1.c.

4.6.1.6 int **parse_and_execute_cmd_lr** (char ∗ *input*, **mips** ∗ *arch*)

Definition at line 566 of file fonctionsStep1.c.

4.7 /home/ben_jah/Bureau/Projet_informatique/step2/include/global.h File Reference

Defines

- #define TAILLESEGMENT 4096
- #define TAILLEMEM TAILLESEGMENT∗256

Typedefs

- typedef unsigned char BYTE

    Definitions (type, constantes...) communes ensemble du projet.
- typedef unsigned int WORD

4.7.1 Define Documentation

4.7.1.1 #define **TAILLEMEM TAILLESEGMENT**∗256

Definition at line 39 of file global.h.

4.7.1.2 #define **TAILLESEGMENT** 4096

Definition at line 35 of file global.h.

4.7.2 Typedef Documentation

4.7.2.1 typedef unsigned char **BYTE**

Definitions (type, constantes...) communes ensemble du projet.

SiMIPS - Simulateur d'un microprocesseur MIPS compatible Linux/ELF

Author

(c) Laurent Fesquet 2003,
(m) Matthieu Chabanas 2005
(m) Frans Portet 2013 <francois.portet@imag.fr>

Definition at line 23 of file global.h.

4.7.2.2 typedef unsigned int WORD

Definition at line 26 of file global.h.

4.8 /home/ben_jah/Bureau/Projet_informatique/step2/include/libelf.h File Reference

`#include <sys/types.h> #include <elf.h>`

Data Structures

- struct Elf_Data
- struct Elf_Arhdr
- struct Elf_Arsym

Defines

- #define ELF_F_DIRTY ELF_F_DIRTY
- #define ELF_F_LAYOUT ELF_F_LAYOUT
- #define ELF_F_PERMISSIVE ELF_F_PERMISSIVE

Typedefs

- typedef struct Elf Elf
- typedef struct Elf_Scn Elf_Scn

Enumerations

- enum Elf_Type { ELF_T_BYTE, ELF_T_ADDR, ELF_T_DYN, ELF_T_EHDR, ELF_T_HALF, ELF_T_OFF, ELF_T_PHDR, ELF_T_RELA, ELF_T_REL, ELF_T_SHDR, ELF_T_SWORD, ELF_T_SYM, ELF_T_WORD, ELF_T_XWORD, ELF_T_SXWORD, ELF_T_VDEF, ELF_T_VDAUX, ELF_T_VNEED, ELF_T_VNAUX, ELF_T_NHDR, ELF_T_SYMINFO, ELF_T_MOVE, ELF_T_LIB, ELF_T_GNUHASH, ELF_T_AUXV, ELF_T_NUM }

- enum Elf_Cmd { ELF_C_NULL, ELF_C_READ, ELF_C_RDWR, ELF_C_WRITE, ELF_C_CLR, ELF_C_SET, ELF_C_FDDONE, ELF_C_FDREAD, ELF_C_READ_MMAP, ELF_C_RDWR_MMAP, ELF_C_WRITE_MMAP, ELF_C_READ_MMAP_PRIVATE, ELF_C_EMPTY, ELF_C_NUM }
- enum { ELF_F_DIRTY = 0x1, ELF_F_DIRTY = 0x1 }
- enum Elf_Kind { ELF_K_NONE, ELF_K_AR, ELF_K_COFF, ELF_K_ELF, ELF_K_NUM }

Functions

- Elf ∗ elf_begin (int __fildes, Elf_Cmd __cmd, Elf ∗__ref)
- Elf ∗ elf_clone (Elf ∗__elf, Elf_Cmd __cmd)
- Elf ∗ elf_memory (char ∗__image, size_t __size)
- Elf_Cmd elf_next (Elf ∗__elf)
- int elf_end (Elf ∗__elf)
- loff_t elf_update (Elf ∗__elf, Elf_Cmd __cmd)
- Elf_Kind elf_kind (Elf ∗__elf) __attribute__((__pure__))
- loff_t elf_getbase (Elf ∗__elf)
- char ∗ elf_getident (Elf ∗__elf, size_t ∗__nbytes)
- Elf32_Ehdr ∗ elf32_getehdr (Elf ∗__elf)
- Elf64_Ehdr ∗ elf64_getehdr (Elf ∗__elf)
- Elf32_Ehdr ∗ elf32_newehdr (Elf ∗__elf)
- Elf64_Ehdr ∗ elf64_newehdr (Elf ∗__elf)
- int elf_getphdrnum (Elf ∗__elf, size_t ∗__dst)
- Elf32_Phdr ∗ elf32_getphdr (Elf ∗__elf)
- Elf64_Phdr ∗ elf64_getphdr (Elf ∗__elf)
- Elf32_Phdr ∗ elf32_newphdr (Elf ∗__elf, size_t __cnt)
- Elf64_Phdr ∗ elf64_newphdr (Elf ∗__elf, size_t __cnt)
- Elf_Scn ∗ elf_getscn (Elf ∗__elf, size_t __index)
- Elf_Scn ∗ elf32_offscn (Elf ∗__elf, Elf32_Off __offset)
- Elf_Scn ∗ elf64_offscn (Elf ∗__elf, Elf64_Off __offset)
- size_t elf_ndxscn (Elf_Scn ∗__scn)
- Elf_Scn ∗ elf_nextscn (Elf ∗__elf, Elf_Scn ∗__scn)
- Elf_Scn ∗ elf_newscn (Elf ∗__elf)
- int elf_scnshndx (Elf_Scn ∗__scn)
- int elf_getshdrnum (Elf ∗__elf, size_t ∗__dst)
- int elf_getshnum (Elf ∗__elf, size_t ∗__dst) __attribute__((__deprecated__))
- int elf_getshdrstrndx (Elf ∗__elf, size_t ∗__dst)
- int elf_getshstrndx (Elf ∗__elf, size_t ∗__dst) __attribute__((__deprecated__))
- Elf32_Shdr ∗ elf32_getshdr (Elf_Scn ∗__scn)
- Elf64_Shdr ∗ elf64_getshdr (Elf_Scn ∗__scn)
- unsigned int elf_flagelf (Elf ∗__elf, Elf_Cmd __cmd, unsigned int __flags)
- unsigned int elf_flagehdr (Elf ∗__elf, Elf_Cmd __cmd, unsigned int __flags)
- unsigned int elf_flagphdr (Elf ∗__elf, Elf_Cmd __cmd, unsigned int __flags)
- unsigned int elf_flagscn (Elf_Scn ∗__scn, Elf_Cmd __cmd, unsigned int __flags)

- unsigned int elf_flagdata (Elf_Data ∗__data, Elf_Cmd __cmd, unsigned int __flags)
- unsigned int elf_flagshdr (Elf_Scn ∗__scn, Elf_Cmd __cmd, unsigned int __flags)
- Elf_Data ∗ elf_getdata (Elf_Scn ∗__scn, Elf_Data ∗__data)
- Elf_Data ∗ elf_rawdata (Elf_Scn ∗__scn, Elf_Data ∗__data)
- Elf_Data ∗ elf_newdata (Elf_Scn ∗__scn)
- Elf_Data ∗ elf_getdata_rawchunk (Elf ∗__elf, loff_t __offset, size_t __size, Elf_Type __type)
- char ∗ elf_strptr (Elf ∗__elf, size_t __index, size_t __offset)
- Elf_Arhdr ∗ elf_getarhdr (Elf ∗__elf)
- loff_t elf_getaroff (Elf ∗__elf)
- size_t elf_rand (Elf ∗__elf, size_t __offset)
- Elf_Arsym ∗ elf_getarsym (Elf ∗__elf, size_t ∗__narsyms)
- int elf_cntl (Elf ∗__elf, Elf_Cmd __cmd)
- char ∗ elf_rawfile (Elf ∗__elf, size_t ∗__nbytes)
- size_t elf32_fsize (Elf_Type __type, size_t __count, unsigned int __version) __attribute__((__const__))
- size_t elf64_fsize (Elf_Type __type, size_t __count, unsigned int __version) __attribute__((__const__))
- Elf_Data ∗ elf32_xlatetom (Elf_Data ∗__dest, const Elf_Data ∗__src, unsigned int __encode)
- Elf_Data ∗ elf64_xlatetom (Elf_Data ∗__dest, const Elf_Data ∗__src, unsigned int __encode)
- Elf_Data ∗ elf32_xlatetof (Elf_Data ∗__dest, const Elf_Data ∗__src, unsigned int __encode)
- Elf_Data ∗ elf64_xlatetof (Elf_Data ∗__dest, const Elf_Data ∗__src, unsigned int __encode)
- int elf_errno (void)
- const char ∗ elf_errmsg (int __error)
- unsigned int elf_version (unsigned int __version)
- void elf_fill (int __fill)
- unsigned long int elf_hash (const char ∗__string) __attribute__((__pure__))
- unsigned long int elf_gnu_hash (const char ∗__string) __attribute__((__pure__))
- long int elf32_checksum (Elf ∗__elf)
- long int elf64_checksum (Elf ∗__elf)

4.8.1 Define Documentation

4.8.1.1 #define ELF_F_DIRTY ELF_F_DIRTY

Definition at line 132 of file libelf.h.

4.8.1.2 #define ELF_F_LAYOUT ELF_F_LAYOUT

Definition at line 134 of file libelf.h.

4.8.1.3 #define ELF_F_PERMISSIVE ELF_F_PERMISSIVE

4.8.2 Typedef Documentation

4.8.2.1 typedef struct Elf Elf

Definition at line 175 of file libelf.h.

4.8.2.2 typedef struct Elf_Scn Elf_Scn

Definition at line 178 of file libelf.h.

4.8.3 Enumeration Type Documentation

4.8.3.1 anonymous enum

Enumerator:

ELF_F_DIRTY

ELF_F_DIRTY

Definition at line 129 of file libelf.h.

4.8.3.2 enum Elf_Cmd

Enumerator:

ELF_C_NULL

ELF_C_READ

ELF_C_RDWR

ELF_C_WRITE

ELF_C_CLR

ELF_C_SET

ELF_C_FDDONE

ELF_C_FDREAD

ELF_C_READ_MMAP

ELF_C_RDWR_MMAP

ELF_C_WRITE_MMAP

ELF_C_READ_MMAP_PRIVATE

ELF_C_EMPTY

ELF_C_NUM

Definition at line 104 of file libelf.h.

4.8.3.3 enum Elf_Kind

Enumerator:

ELF_K_NONE

ELF_K_AR

ELF_K_COFF

ELF_K_ELF

ELF_K_NUM

Definition at line 141 of file libelf.h.

4.8.3.4 enum Elf_Type

Enumerator:

ELF_T_BYTE

ELF_T_ADDR

ELF_T_DYN

ELF_T_EHDR

ELF_T_HALF

ELF_T_OFF

ELF_T_PHDR

ELF_T_RELA

ELF_T_REL

ELF_T_SHDR

ELF_T_SWORD

ELF_T_SYM

ELF_T_WORD

ELF_T_XWORD

ELF_T_SXWORD

ELF_T_VDEF

ELF_T_VDAUX

ELF_T_VNEED

ELF_T_VNAUX

ELF_T_NHDR

ELF_T_SYMINFO

ELF_T_MOVE

ELF_T_LIB

ELF_T_GNUHASH

ELF_T_AUXV

ELF_T_NUM

Definition at line 60 of file libelf.h.

4.8.4 Function Documentation

4.8.4.1 long int elf32_checksum (Elf ∗ __elf)

4.8.4.2 size_t elf32_fsize (Elf_Type __type, size_t __count, unsigned int __version) const

4.8.4.3 Elf32_Ehdr∗ elf32_getehdr (Elf ∗ __elf)

4.8.4.4 Elf32_Phdr∗ elf32_getphdr (Elf ∗ __elf)

4.8.4.5 Elf32_Shdr∗ elf32_getshdr (Elf_Scn ∗ __scn)

4.8.4.6 Elf32_Ehdr∗ elf32_newehdr (Elf ∗ __elf)

4.8.4.7 Elf32_Phdr∗ elf32_newphdr (Elf ∗ __elf, size_t __cnt)

4.8.4.8 Elf_Scn∗ elf32_offscn (Elf ∗ __elf, Elf32_Off __offset)

4.8.4.9 Elf_Data∗ elf32_xlatetof (Elf_Data ∗ __dest, const Elf_Data ∗ __src, unsigned int __encode)

4.8.4.10 Elf_Data∗ elf32_xlatetom (Elf_Data ∗ __dest, const Elf_Data ∗ __src, unsigned int __encode)

4.8.4.11 long int elf64_checksum (Elf ∗ __elf)

4.8.4.12 size_t elf64_fsize (Elf_Type __type, size_t __count, unsigned int __version) const

4.8.4.13 Elf64_Ehdr∗ elf64_getehdr (Elf ∗ __elf)

4.8.4.14 Elf64_Phdr∗ elf64_getphdr (Elf ∗ __elf)

4.8.4.15 Elf64_Shdr∗ elf64_getshdr (Elf_Scn ∗ __scn)

4.8.4.16 Elf64_Ehdr∗ elf64_newehdr (Elf ∗ __elf)

4.8.4.17 Elf64_Phdr∗ elf64_newphdr (Elf ∗ __elf, size_t __cnt)

4.8.4.18 Elf_Scn∗ elf64_offscn (Elf ∗ __elf, Elf64_Off __offset)

4.8.4.19 Elf_Data∗ elf64_xlatetof (Elf_Data ∗ __dest, const Elf_Data ∗ __src, unsigned int __encode)

4.8.4.20 Elf_Data∗ elf64_xlatetom (Elf_Data ∗ __dest, const Elf_Data ∗ __src, unsigned int __encode)

4.8.4.21 Elf∗ elf_begin (int __fildes, Elf_Cmd __cmd, Elf ∗ __ref)

4.8.4.22 Elf∗ elf_clone (Elf ∗ *__elf,* Elf_Cmd *__cmd*)

4.8.4.23 int elf_cntl (Elf ∗ *__elf,* Elf_Cmd *__cmd*)

4.8.4.24 int elf_end (Elf ∗ *__elf*)

4.8.4.25 const char∗ elf_errmsg (int *__error*)

4.8.4.26 int elf_errno (void)

4.8.4.27 void elf_fill (int *__fill*)

4.8.4.28 unsigned int elf_flagdata (Elf_Data ∗ *__data,* Elf_Cmd *__cmd,* unsigned int *__flags*)

4.8.4.29 unsigned int elf_flagehdr (Elf ∗ *__elf,* Elf_Cmd *__cmd,* unsigned int *__flags*)

4.8.4.30 unsigned int elf_flagelf (Elf ∗ *__elf,* Elf_Cmd *__cmd,* unsigned int *__flags*)

4.8.4.31 unsigned int elf_flagphdr (Elf ∗ *__elf,* Elf_Cmd *__cmd,* unsigned int *__flags*)

4.8.4.32 unsigned int elf_flagscn (Elf_Scn ∗ *__scn,* Elf_Cmd *__cmd,* unsigned int *__flags*)

4.8.4.33 unsigned int elf_flagshdr (Elf_Scn ∗ *__scn,* Elf_Cmd *__cmd,* unsigned int *__flags*)

4.8.4.34 Elf_Arhdr∗ elf_getarhdr (Elf ∗ *__elf*)

4.8.4.35 loff_t elf_getaroff (Elf ∗ *__elf*)

4.8.4.36 Elf_Arsym∗ elf_getarsym (Elf ∗ *__elf,* size_t ∗ *__narsyms*)

4.8.4.37 loff_t elf_getbase (Elf ∗ *__elf*)

4.8.4.38 Elf_Data∗ elf_getdata (Elf_Scn ∗ *__scn,* Elf_Data ∗ *__data*)

4.8.4.39 Elf_Data∗ elf_getdata_rawchunk (Elf ∗ *__elf,* loff_t *__offset,* size_t *__size,* Elf_Type *__type*)

4.8.4.40 char∗ elf_getident (Elf ∗ *__elf,* size_t ∗ *__nbytes*)

4.8.4.41 int elf_getphdrnum (Elf ∗ *__elf,* size_t ∗ *__dst*)

4.8.4.42 Elf_Scn∗ elf_getscn (Elf ∗ *__elf,* size_t *__index*)

4.8.4.43 int elf_getshdrnum (Elf ∗ *__elf,* size_t ∗ *__dst*)

4.8.4.44 int elf_getshdrstrndx (Elf ∗ __elf, size_t ∗ __dst)

4.8.4.45 int elf_getshnum (Elf ∗ __elf, size_t ∗ __dst)

4.8.4.46 int elf_getshstrndx (Elf ∗ __elf, size_t ∗ __dst)

4.8.4.47 unsigned long int elf_gnu_hash (const char ∗ __string)

4.8.4.48 unsigned long int elf_hash (const char ∗ __string)

4.8.4.49 Elf_Kind elf_kind (Elf ∗ __elf)

4.8.4.50 Elf∗ elf_memory (char ∗ __image, size_t __size)

4.8.4.51 size_t elf_ndxscn (Elf_Scn ∗ __scn)

4.8.4.52 Elf_Data∗ elf_newdata (Elf_Scn ∗ __scn)

4.8.4.53 Elf_Scn∗ elf_newscn (Elf ∗ __elf)

4.8.4.54 Elf_Cmd elf_next (Elf ∗ __elf)

4.8.4.55 Elf_Scn∗ elf_nextscn (Elf ∗ __elf, Elf_Scn ∗ __scn)

4.8.4.56 size_t elf_rand (Elf ∗ __elf, size_t __offset)

4.8.4.57 Elf_Data∗ elf_rawdata (Elf_Scn ∗ __scn, Elf_Data ∗ __data)

4.8.4.58 char∗ elf_rawfile (Elf ∗ __elf, size_t ∗ __nbytes)

4.8.4.59 int elf_scnshndx (Elf_Scn ∗ __scn)

4.8.4.60 char∗ elf_strptr (Elf ∗ __elf, size_t __index, size_t __offset)

4.8.4.61 loff_t elf_update (Elf ∗ __elf, Elf_Cmd __cmd)

4.8.4.62 unsigned int elf_version (unsigned int __version)

4.9 /home/ben_jah/Bureau/Projet_informatique/step2/include/man.h File Reference

Functions

- int man (void)

4.9.1 Function Documentation

4.9.1.1 int man (void)

Definition at line 25 of file man.c.

4.10 /home/ben_jah/Bureau/Projet_informatique/step2/include/mipself.h File Reference

interface du chargement en memoire d'un fichier ELF.

```
#include "elfimport.h" #include "global.h" #include <libelf.-
h>
```

Data Structures

- struct SectionELF

  Type de donn SectionELF ! Definition des informations d'une section ELF : (nom, donn, taille des donn, adresse mire de la section, ...) Cette structure d'ange, quasi-ivalente structure MemZone dnie dans mipsloader.c, est indndante de la librairie ELF. Elle est est utilisour les interfaces avec le shell!
- struct MemZone

  Type de donn zone ! Definition des informations d'une section ELF : (nom, donn, taille des donn, adresse mire de la section, ...) + acces aux differentes sections du fichier ELF. Cette structure de donn est completement dndante de la librairie ELF.

Defines

- #define NUMZONE 3 /∗ Nombre de zones: text, data et bss ∗/

Functions

- void printELFSection (SectionELF ∗section)
- char ∗ relocName (unsigned char type)
- char ∗ shtName (Elf32_Word SectionType)
- char ∗ stName (unsigned char t)
- char ∗ getSectionHeaderName (Elf32_Word index)
- char ∗ getName (Elf32_Word index)
- Elf32_Sym ∗ getSymb (Elf32_Word index)
- void freeHashTable ()
- int hashCode (unsigned long adr)
- void addSymbol (unsigned long adr, char ∗ident)
- char ∗ getAddressName (unsigned long adr)
- int mipsloader (const char ∗filename, SectionELF ∗textSection, SectionELF ∗dataSection, SectionELF ∗bssSection)

4.10.1 Detailed Description

interface du chargement en memoire d'un fichier ELF.

Author

(c) Laurent Fesquet 2003,
(m) Matthieu Chabanas 2005
(m) Frans Portet 2013 <francois.portet@imag.fr>

interface du chargement en memoire d'un fichier ELF.

Definition in file mipself.h.

4.10.2 Define Documentation

4.10.2.1 #define **NUMZONE** 3 /* Nombre de zones: text, data et bss */

Definition at line 19 of file mipself.h.

4.10.3 Function Documentation

4.10.3.1 void **addSymbol** (unsigned long *adr,* char * *ident*)

Definition at line 318 of file mipself.c.

4.10.3.2 void **freeHashTable** ( )

Definition at line 285 of file mipself.c.

4.10.3.3 char* **getAddressName** (unsigned long *adr*)

Definition at line 345 of file mipself.c.

4.10.3.4 char* **getName** (Elf32_Word *index*)

Definition at line 227 of file mipself.c.

4.10.3.5 char* **getSectionHeaderName** (Elf32_Word *index*)

Definition at line 210 of file mipself.c.

4.10.3.6 Elf32_Sym* **getSymb** (Elf32_Word *index*)

Definition at line 244 of file mipself.c.

4.10.3.7 int hashCode (unsigned long *adr*)

Definition at line 303 of file mipself.c.

4.10.3.8 int mipsloader (const char * *filename,* SectionELF * *textSection,* SectionELF * *dataSection,* SectionELF * *bssSection*)

Definition at line 551 of file mipself.c.

4.10.3.9 void printELFSection (SectionELF * *section*)

Definition at line 364 of file mipself.c.

4.10.3.10 char* relocName (unsigned char *type*)

Definition at line 79 of file mipself.c.

4.10.3.11 char* shtName (Elf32_Word *SectionType*)

Definition at line 124 of file mipself.c.

4.10.3.12 char* stName (unsigned char *t*)

Definition at line 176 of file mipself.c.

4.11 /home/ben_jah/Bureau/Projet_informatique/step2/include/notify.h File Reference

Notification management.

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

Defines

- #define FOR_ERRORS 0x00
- #define FOR_WARNINGS 0x01
- #define FOR_INFOS 0x02
- #define STYLE_OFF 0x00
- #define STYLE_BOLD 0x01
- #define STYLE_USCORE 0x04
- #define STYLE_BLINK 0x05
- #define STYLE_REVERSE 0x07

- #define STYLE_CONCEALED 0x08
- #define COLOR_BLACK 30
- #define COLOR_RED 31
- #define COLOR_GREEN 32
- #define COLOR_YELLOW 33
- #define COLOR_BLUE 34
- #define COLOR_MAGENTA 35
- #define COLOR_CYAN 36
- #define COLOR_WHITE 37
- #define STYLE_ERROR STYLE_BLINK
- #define STYLE_WARNING STYLE_BOLD
- #define STYLE_INFO STYLE_BOLD
- #define COLOR_ERROR COLOR_RED
- #define COLOR_WARNING COLOR_YELLOW
- #define COLOR_INFO COLOR_GREEN
- #define STYLE(purpose)
- #define COLOR(purpose)
- #define ON(stream) stream
- #define RESET_COLORS(on_stream) fprintf(on_stream, "%c[%dm", 0x1B, 0)
- #define SET_COLORS(purpose, on_stream)
- #define ERROR_MSG(...)
- #define WARNING_MSG(...)
- #define INFO_MSG(...)
- #define DEBUG_MSG(...)

4.11.1 Detailed Description

Notification management.

Author

François Cayre <cayre.(null)>

Date

Thu Mar 22 17:22:55 2012 Notification management.

Definition in file notify.h.

4.11.2 Define Documentation

4.11.2.1 #define COLOR(*purpose*)

Value:

```
(purpose == FOR_ERRORS ? COLOR_ERROR :                    \
   (purpose == FOR_WARNINGS ? COLOR_WARNING : COLOR_INFO ))
```

Definition at line 53 of file notify.h.

4.11.2.2 #define COLOR_BLACK 30

Definition at line 32 of file notify.h.

4.11.2.3 #define COLOR_BLUE 34

Definition at line 36 of file notify.h.

4.11.2.4 #define COLOR_CYAN 36

Definition at line 38 of file notify.h.

4.11.2.5 #define COLOR_ERROR COLOR_RED

Definition at line 45 of file notify.h.

4.11.2.6 #define COLOR_GREEN 32

Definition at line 34 of file notify.h.

4.11.2.7 #define COLOR_INFO COLOR_GREEN

Definition at line 47 of file notify.h.

4.11.2.8 #define COLOR_MAGENTA 35

Definition at line 37 of file notify.h.

4.11.2.9 #define COLOR_RED 31

Definition at line 33 of file notify.h.

4.11.2.10 #define COLOR_WARNING COLOR_YELLOW

Definition at line 46 of file notify.h.

4.11.2.11 #define COLOR_WHITE 37

Definition at line 39 of file notify.h.

4.11.2.12 #define COLOR_YELLOW 33

Definition at line 35 of file notify.h.

4.11.2.13 #define DEBUG_MSG(*...*)

Definition at line 110 of file notify.h.

4.11.2.14 #define ERROR_MSG(*...*)

Value:

```
do {                                                    \
    fprintf( stderr, "%c[%d;%dm", 0x1B, STYLE_BOLD, COLOR_BLUE );       \
    fprintf( stderr, "[ ERROR :: %s:%s:%d] ",                           \
             __FILE__, __FUNCTION__, __LINE__ );                        \
    SET_COLORS(FOR_ERRORS, ON(stderr));                                 \
    fprintf( stderr, __VA_ARGS__ );                                     \
    fprintf( stderr, ".\n" );                                           \
    RESET_COLORS(ON(stderr));                                           \
    exit( EXIT_FAILURE );                                               \
} while( 0 )
```

Definition at line 64 of file notify.h.

4.11.2.15 #define FOR_ERRORS 0x00

Definition at line 21 of file notify.h.

4.11.2.16 #define FOR_INFOS 0x02

Definition at line 23 of file notify.h.

4.11.2.17 #define FOR_WARNINGS 0x01

Definition at line 22 of file notify.h.

4.11.2.18 #define INFO_MSG(*...*)

Definition at line 96 of file notify.h.

4.11.2.19 #define ON(*stream*) stream

Definition at line 57 of file notify.h.

4.11.2.20 #define RESET_COLORS(*on_stream*) fprintf(on_stream, ”%c[%dm”, 0x1B, 0)

Definition at line 58 of file notify.h.

4.11.2.21 #define SET_COLORS(*purpose*, *on_stream*)

Value:

```
fprintf( on_stream, "%c[%d;%dm", 0x1B,  \
           STYLE(purpose), COLOR(purpose) )
```

Definition at line 60 of file notify.h.

4.11.2.22 #define STYLE(*purpose*)

Value:

```
(purpose == FOR_ERRORS ? STYLE_ERROR :                  \
   (purpose == FOR_WARNINGS ? STYLE_WARNING : STYLE_INFO ))
```

Definition at line 49 of file notify.h.

4.11.2.23 #define STYLE_BLINK 0x05

Definition at line 28 of file notify.h.

4.11.2.24 #define STYLE_BOLD 0x01

Definition at line 26 of file notify.h.

4.11.2.25 #define STYLE_CONCEALED 0x08

Definition at line 30 of file notify.h.

4.11.2.26 #define STYLE_ERROR STYLE_BLINK

Definition at line 41 of file notify.h.

4.11.2.27 #define STYLE_INFO STYLE_BOLD

Definition at line 43 of file notify.h.

4.11.2.28 #define STYLE_OFF 0x00

Definition at line 25 of file notify.h.

4.11.2.29 #define STYLE_REVERSE 0x07

Definition at line 29 of file notify.h.

4.11.2.30 #define STYLE_USCORE 0x04

Definition at line 27 of file notify.h.

4.11.2.31 #define STYLE_WARNING STYLE_BOLD

Definition at line 42 of file notify.h.

4.11.2.32 #define WARNING_MSG(...)

Value:

```
do {                                            \
    fprintf( stderr, "%c[%d;%dm", 0x1B, STYLE_BOLD, COLOR_BLUE );       \
    fprintf( stderr, "[WARNING:: %s:%s:%d] ",                           \
             __FILE__, __FUNCTION__, __LINE__ );                        \
    SET_COLORS(FOR_WARNINGS, ON(stderr));                               \
    fprintf( stderr, __VA_ARGS__ );                                     \
    fprintf( stderr, ".\n" );                                           \
    RESET_COLORS(ON(stderr));                                           \
} while( 0 )
```

Definition at line 75 of file notify.h.

4.12 /home/ben_jah/Bureau/Projet_informatique/step2/include/structure.h File Reference

Data Structures

- struct registre
- struct mips

Defines

- #define DATA 0
- #define TEXT 1
- #define BSS 2

4.12.1 Define Documentation

4.12.1.1 #define BSS 2

Definition at line 6 of file structure.h.

4.12.1.2 #define DATA 0

Definition at line 4 of file structure.h.

4.12.1.3 #define TEXT 1

Definition at line 5 of file structure.h.

4.13 /home/ben_jah/Bureau/Projet_informatique/step2/include/testsStep1.h File Reference

Functions

- int test_registre (char ∗input)
- int test_memoire (mips ∗arch, unsigned int adresse, unsigned int ∗addr_in_block)
- int test_valeur (unsigned int val)

4.13.1 Function Documentation

4.13.1.1 int test_memoire (mips ∗ *arch*, unsigned int *adresse*, unsigned int ∗ *addr_in_block*)

Definition at line 130 of file testsStep1.c.

4.13.1.2 int test_registre (char ∗ *input*)

Definition at line 26 of file testsStep1.c.

4.13.1.3 int test_valeur (unsigned int *val*)

Definition at line 152 of file testsStep1.c.

4.14 /home/ben_jah/Bureau/Projet_informatique/step2/src/cmdSearch.c File Reference

#include <stdlib.h> #include <stdio.h> #include <string.h> #include <libelf.h> #include <math.h> #include <readline/readline.h> #include <readline/history.h> #include "global.h" #include "mipself.h" #include "notify.h" #include "structure.h" #include "constantes.h" #include "fonctions.h" #include "fonctionsStep1.h" #include "testsStep1.h" #include "fonctionsStep2.h" #include "elfimport.h" #include "cmdSearch.h"

Functions

- char ∗ cmdSearch (char ∗type, unsigned int param)

4.14.1 Function Documentation

4.14.1.1 char∗ cmdSearch (char ∗ *type*, unsigned int *param*)

Definition at line 27 of file cmdSearch.c.

4.15 /home/ben_jah/Bureau/Projet_informatique/step2/src/fonctions.c File Reference

#include <stdlib.h> #include <stdio.h> #include <string.h> #include <libelf.h> #include <readline/readline.h>× #include <readline/history.h> #include "global.h" #include "mipself.h" #include "notify.h" #include "structure.h"× #include "constantes.h" #include "fonctions.h" #include "fonctionsStep1.h" #include "testsStep1.h" #include "fonctionsStep2.h" #include "elfimport.h"

Functions

- int execute_cmd_testcmd (int hexValue)
- int parse_and_execute_cmd_testcmd (char ∗paramsStr, mips ∗arch)
- int execute_cmd_exit ()
- int parse_and_execute_cmd_exit (char ∗paramsStr)
- int parse_and_execute_cmd_string (char ∗input, mips ∗arch)
- int acquire_line (FILE ∗fp, char ∗input)
- void usage_ERROR_MSG (char ∗command)

4.15.1 Function Documentation

4.15.1.1 int acquire_line (FILE ∗ *fp,* char ∗ *input*)

Definition at line 146 of file fonctions.c.

4.15.1.2 int execute_cmd_exit ( )

Definition at line 59 of file fonctions.c.

4.15.1.3 int execute_cmd_testcmd (int *hexValue*)

Definition at line 30 of file fonctions.c.

4.15.1.4 int parse_and_execute_cmd_exit (char ∗ *paramsStr*)

Definition at line 66 of file fonctions.c.

4.15.1.5 int parse_and_execute_cmd_string (char ∗ *input,* mips ∗ *arch*)

Definition at line 79 of file fonctions.c.

4.15.1.6 int parse_and_execute_cmd_testcmd (char ∗ *paramsStr,* mips ∗ *arch*)

Definition at line 43 of file fonctions.c.

4.15.1.7 void usage_ERROR_MSG (char ∗ *command*)

Definition at line 178 of file fonctions.c.

4.16 /home/ben_jah/Bureau/Projet_informatique/step2/src/fonctionsStep1.c File Reference

```
#include <stdlib.h> #include <stdio.h> #include <string.h> #include <libelf.h> #include <readline/readline.h> ×
#include <readline/history.h> #include "global.h" #include "mipself.h" #include "notify.h" #include "structure.h" ×
#include "constantes.h" #include "fonctions.h" #include "fonctionsStep1.h" #include "testsStep1.h" #include "fonctionsStep2.h" #include "elfimport.h"
```

Functions

- void init_mips (mips ∗arch)
- void init_segment (SectionELF ∗seg, int type)
- int execute_cmd_lm (unsigned int adr, unsigned int val, mips ∗arch)
- int parse_and_execute_cmd_lm (char ∗paramsStr, mips ∗arch)
- int execute_cmd_dm (int cas, unsigned int addr, unsigned int arg2, mips ∗arch)
- int parse_and_execute_cmd_dm (char ∗paramsStr, mips ∗arch)
- int execute_cmd_dr (int test_reg, mips ∗arch)
- int parse_and_execute_cmd_dr (char ∗input, mips ∗arch)
- int execute_cmd_lr (int adr, unsigned int val, mips ∗arch)
- int parse_and_execute_cmd_lr (char ∗input, mips ∗arch)

4.16.1 Function Documentation

4.16.1.1 int **execute_cmd_dm** (int *cas,* unsigned int *addr,* unsigned int *arg2,* **mips** ∗ *arch*)

Definition at line 257 of file fonctionsStep1.c.

4.16.1.2 int **execute_cmd_dr** (int *test_reg,* **mips** ∗ *arch*)

Definition at line 488 of file fonctionsStep1.c.

4.16.1.3 int **execute_cmd_lm** (unsigned int *adr,* unsigned int *val,* **mips** ∗ *arch*)

Definition at line 135 of file fonctionsStep1.c.

4.16.1.4 int **execute_cmd_lr** (int *adr,* unsigned int *val,* **mips** ∗ *arch*)

Definition at line 550 of file fonctionsStep1.c.

4.16.1.5 void **init_mips** (**mips** ∗ *arch*)

Definition at line 39 of file fonctionsStep1.c.

4.16.1.6 void **init_segment** (**SectionELF** ∗ *seg,* int *type*)

Definition at line 87 of file fonctionsStep1.c.

4.16.1.7 int **parse_and_execute_cmd_dm** (char ∗ *paramsStr,* **mips** ∗ *arch*)

Definition at line 381 of file fonctionsStep1.c.

4.16.1.8 int parse_and_execute_cmd_dr (char ∗ *input*, mips ∗ *arch*)

Definition at line 519 of file fonctionsStep1.c.

4.16.1.9 int parse_and_execute_cmd_lm (char ∗ *paramsStr*, mips ∗ *arch*)

Definition at line 164 of file fonctionsStep1.c.

4.16.1.10 int parse_and_execute_cmd_lr (char ∗ *input*, mips ∗ *arch*)

Definition at line 566 of file fonctionsStep1.c.

4.17 /home/ben_jah/Bureau/Projet_informatique/step2/src/fonctionsStep2.c File Reference

```
#include <stdlib.h> #include <stdint.h> #include <stdio.h> #include <string.h> #include <libelf.h> #include <math.h> #include <readline/readline.h> #include <readline/history.h> #include "global.h" #include "mipself.h" #include "notify.h" #include "structure.h" #include "constantes.h" #include "fonctions.h" #include "fonctionsStep1.h" #include "testsStep1.h" #include "fonctionsStep2.h" #include "elfimport.h"
```

Functions

- unsigned int getbits (unsigned int val, unsigned int start, unsigned int stop)
- int parse_and_execute_cmd_da (char ∗input, mips ∗arch)
- int execute_cmd_da (unsigned int addr, unsigned int nbr_instr, mips ∗arch)
- int parse_and_execute_cmd_lp (char ∗input, mips ∗arch)
- int execute_cmd_lp (FILE ∗program, char ∗nom_fichier, mips ∗arch)

4.17.1 Function Documentation

4.17.1.1 int execute_cmd_da (unsigned int *addr*, unsigned int *nbr_instr*, mips ∗ *arch*)

Definition at line 141 of file fonctionsStep2.c.

4.17.1.2 int execute_cmd_lp (FILE ∗ *program*, char ∗ *nom_fichier*, mips ∗ *arch*)

Definition at line 339 of file fonctionsStep2.c.

4.17.1.3 unsigned int **getbits** (unsigned int *val,* unsigned int *start,* unsigned int *stop*)

Definition at line 26 of file fonctionsStep2.c.

4.17.1.4 int **parse_and_execute_cmd_da** (char ∗ *input,* **mips** ∗ *arch*)

Definition at line 55 of file fonctionsStep2.c.

4.17.1.5 int **parse_and_execute_cmd_lp** (char ∗ *input,* **mips** ∗ *arch*)

Definition at line 291 of file fonctionsStep2.c.

4.18 /home/ben_jah/Bureau/Projet_informatique/step2/src/man.c File Reference

`#include <stdlib.h> #include <stdio.h> #include <string.h> #include <libelf.h> #include <readline/readline.h>` × `#include <readline/history.h> #include "global.h" #include "mipself.h" #include "notify.h" #include "structure.h"` × `#include "constantes.h" #include "fonctions.h" #include "fonctionsStep1.h" #include "testsStep1.h" #include "fonctionsStep2.h" #include "elfimport.h" #include "man.h"`

Functions

- int man (void)

4.18.1 Function Documentation

4.18.1.1 int **man** (void)

Definition at line 25 of file man.c.

4.19 /home/ben_jah/Bureau/Projet_informatique/step2/src/mipself.c File Reference

`#include <stdio.h> #include <stdlib.h> #include <string.h> #include <fcntl.h> #include <unistd.h> #include <libelf.h> #include <readline/readline.h> #include <readline/history.h> #include "global.h" #include "mipself.h" #include "notify.h" #include "structure.h" #include "constantes.h" #include`

```
"fonctions.h" #include "fonctionsStep1.h" #include "tests-
Step1.h" #include "fonctionsStep2.h" #include "elfimport.-
h"
```

Data Structures

- struct nodeSymbol

Defines

- #define HASHSIZE 37

Typedefs

- typedef struct nodeSymbol ∗ listSymboles

Functions

- char ∗ relocName (unsigned char type)
- char ∗ shtName (Elf32_Word SectionType)
- char ∗ stName (unsigned char t)
- char ∗ getSectionHeaderName (Elf32_Word index)
- char ∗ getName (Elf32_Word index)
- Elf32_Sym ∗ getSymb (Elf32_Word index)
- void freeHashTable ()
- int hashCode (unsigned long adr)
- void addSymbol (unsigned long adr, char ∗ident)
- char ∗ getAddressName (unsigned long adr)
- void printELFSection (SectionELF ∗section)
- int mipsloader (const char ∗filename, SectionELF ∗textSection, SectionELF ∗dataSection, SectionELF ∗bssSection)

4.19.1 Define Documentation

4.19.1.1 #define HASHSIZE 37

Definition at line 262 of file mipself.c.

4.19.2 Typedef Documentation

4.19.2.1 typedef struct nodeSymbol∗ listSymboles

Definition at line 272 of file mipself.c.

4.19.3 Function Documentation

4.19.3.1 void addSymbol (unsigned long *adr*, char ∗ *ident*)

Definition at line 318 of file mipself.c.

4.19.3.2 void freeHashTable ( )

Definition at line 285 of file mipself.c.

4.19.3.3 char∗ getAddressName (unsigned long *adr*)

Definition at line 345 of file mipself.c.

4.19.3.4 char∗ getName (Elf32_Word *index*)

Definition at line 227 of file mipself.c.

4.19.3.5 char∗ getSectionHeaderName (Elf32_Word *index*)

Definition at line 210 of file mipself.c.

4.19.3.6 Elf32_Sym∗ getSymb (Elf32_Word *index*)

Definition at line 244 of file mipself.c.

4.19.3.7 int hashCode (unsigned long *adr*)

Definition at line 303 of file mipself.c.

4.19.3.8 int mipsloader (const char ∗ *filename*, SectionELF ∗ *textSection*, SectionELF ∗ *dataSection*, SectionELF ∗ *bssSection*)

Definition at line 551 of file mipself.c.

4.19.3.9 void printELFSection (SectionELF ∗ *section*)

Definition at line 364 of file mipself.c.

4.19.3.10 char∗ relocName (unsigned char *type*)

Definition at line 79 of file mipself.c.

4.19.3.11 char∗ **shtName** (Elf32_Word *SectionType*)

Definition at line 124 of file mipself.c.

4.19.3.12 char∗ **stName** (unsigned char *t*)

Definition at line 176 of file mipself.c.

4.20 /home/ben_jah/Bureau/Projet_informatique/step2/src/simMips.c File Reference

Main pour le début du projet simulateur MIPS.

`#include <stdlib.h> #include <stdio.h> #include <string.h> #include <libelf.h> #include <readline/readline.h>` × `#include <readline/history.h> #include "global.h" #include "mipself.h" #include "notify.h" #include "structure.h"` × `#include "constantes.h" #include "fonctions.h" #include "fonctionsStep1.h" #include "testsStep1.h" #include "fonctionsStep2.h" #include "elfimport.h" #include "man.h"`

Functions

- int main (int argc, char ∗argv[ ])

4.20.1 Detailed Description

Main pour le début du projet simulateur MIPS.

Author

François Cayre, Nicolas Castagné

Date

Fri Jun 15 18:13:02 2012 puis aout 2013

Definition in file simMips.c.

4.20.2 Function Documentation

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

Definition at line 46 of file simMips.c.

4.21 /home/ben_jah/Bureau/Projet_informatique/step2/src/testsStep1.c File Reference

`#include <stdlib.h> #include <stdio.h> #include <string.h> #include <libelf.h> #include <readline/readline.h>` × `#include <readline/history.h> #include "global.h" #include "mipself.h" #include "notify.h" #include "structure.h"` × `#include "constantes.h" #include "fonctions.h" #include "fonctionsStep1.h" #include "testsStep1.h" #include "fonctionsStep2.h" #include "elfimport.h"`

Functions

- int test_registre (char ∗input)
- int test_memoire (mips ∗arch, unsigned int adresse, unsigned int ∗addr_in_block)
- int test_valeur (unsigned int val)

4.21.1 Function Documentation

4.21.1.1 int **test_memoire** (**mips** ∗ ***arch,*** unsigned int ***adresse,*** unsigned int ∗ ***addr_in_block***)

Definition at line 130 of file testsStep1.c.

4.21.1.2 int **test_registre** (char ∗ ***input***)

Definition at line 26 of file testsStep1.c.

4.21.1.3 int **test_valeur** (unsigned int ***val***)

Definition at line 152 of file testsStep1.c.