

new/usr/src/head/libelf.h

1

```
*****
5176 Fri Apr 4 04:24:02 2014
new/usr/src/head/libelf.h
4555 macro side-effects with /usr/include/libelf.h
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*      Copyright (c) 1988 AT&T */
22 /*      All Rights Reserved      */

24 /*
25  * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
26  * Use is subject to license terms.
27  * Copyright 2014 PALO, Richard. All rights reserved.
28  */

30 #ifndef _LIBELF_H
31 #define _LIBELF_H

33 #include <sys/types.h>
34 #include <sys/elf.h>

37 #ifdef __cplusplus
38 extern "C" {
39 #endif

42 #if defined(_ILP32) && (_FILE_OFFSET_BITS != 32)
43 #error "large files are not supported by libelf"
44 #endif

46 #undef _
47 #ifdef __STDC__
48 typedef void Elf_Void;
49 #define _(a) a
50 #else
51 typedef char Elf_Void;
52 #define _(a) ()
53 #undef const
54 #define const
55 #endif

48 /*
49  * Commands
50  */
```

new/usr/src/head/libelf.h

2

```
51 typedef enum {
52     ELF_C_NULL = 0, /* must be first, 0 */
53     ELF_C_READ,
54     ELF_C_WRITE,
55     ELF_C_CLR,
56     ELF_C_SET,
57     ELF_C_FDDONE,
58     ELF_C_FDREAD,
59     ELF_C_RDWR,
60     ELF_C_WRIMAGE,
61     ELF_C_IMAGE,
62     ELF_C_NUM /* must be last */
63 } Elf_Cmd;
unchanged_portion_omitted

156 /*
157  * Function declarations
158  */
159 Elf *elf_begin(int, Elf_Cmd, Elf *);
160 int elf_cntl(Elf *, Elf_Cmd);
161 int elf_end(Elf *);
162 const char *elf_errmsg(int);
163 int elf_errno(void);
164 void elf_fill(int);
165 unsigned elf_flagdata(Elf_Data *, Elf_Cmd, unsigned);
166 unsigned elf_flagehdr(Elf *, Elf_Cmd, unsigned);
167 unsigned elf_flagelf(Elf *, Elf_Cmd, unsigned);
168 unsigned elf_flagphdr(Elf *, Elf_Cmd, unsigned);
169 unsigned elf_flagscn(Elf_Scn *, Elf_Cmd, unsigned);
170 unsigned elf_flagshdr(Elf_Scn *, Elf_Cmd, unsigned);
171 size_t elf32_fsize(Elf_Type, size_t, unsigned);
172 Elf_Arhdr *elf_getarhdr(Elf *);
173 Elf_Arsym *elf_getarsym(Elf *, size_t *);
174 off_t elf_getbase(Elf *);
175 Elf_Data *elf_getdata(Elf_Scn *, Elf_Data *);
176 Elf32_Ehdr *elf32_getehdr(Elf *);
177 char *elf_getident(Elf *, size_t *);
178 Elf32_Phdr *elf32_getphdr(Elf *);
179 Elf_Scn *elf_getscn(Elf *elf, size_t);
180 Elf32_Shdr *elf32_getshdr(Elf_Scn *);
181 int elf_getphnum(Elf *, size_t *);
182 int elf_getphdrnum(Elf *, size_t *);
183 int elf_getshnum(Elf *, size_t *);
184 int elf_getshdrnum(Elf *, size_t *);
185 int elf_getshstrndx(Elf *, size_t *);
186 int elf_getshdrstrndx(Elf *, size_t *);
187 unsigned long elf_hash(const char *);
188 uint_t elf_sys_encoding(void);
189 long elf32_checksum(Elf *);
190 Elf_Kind elf_kind(Elf *);
191 Elf *elf_memory(char *, size_t);
192 size_t elf_ndxscn(Elf_Scn *);
193 Elf_Data *elf_newdata(Elf_Scn *);
194 Elf32_Ehdr *elf32_newehdr(Elf *);
195 Elf32_Phdr *elf32_newphdr(Elf *, size_t);
196 Elf_Scn *elf_newscn(Elf *);
197 Elf_Scn *elf_nextscn(Elf *, Elf_Scn *);
198 Elf_Cmd elf_next(Elf *);
199 size_t elf_rand(Elf *, size_t);
200 Elf_Data *elf_rawdata(Elf_Scn *, Elf_Data *);
201 char *elf_rawfile(Elf *, size_t *);
202 char *elf_strptr(Elf *, size_t, size_t);
203 off_t elf_update(Elf *, Elf_Cmd);
204 unsigned elf_version(unsigned);
205 Elf_Data *elf32_xlatetof(Elf_Data *, const Elf_Data *, unsigned);
```

```

206 Elf_Data      *elf32_xlatetom(Elf_Data *, const Elf_Data *, unsigned);
169 Elf          *elf_begin      _((int, Elf_Cmd, Elf *));
170 int          elf_cntl        _((Elf *, Elf_Cmd));
171 int          elf_end          _((Elf *));
172 const char   *elf_errmsg     _((int));
173 int          elf_errno       _((void));
174 void         elf_fill        _((int));
175 unsigned     elf_flagdata    _((Elf_Data *, Elf_Cmd, unsigned));
176 unsigned     elf_flagehdr    _((Elf *, Elf_Cmd, unsigned));
177 unsigned     elf_flagelf     _((Elf *, Elf_Cmd, unsigned));
178 unsigned     elf_flagphdr    _((Elf *, Elf_Cmd, unsigned));
179 unsigned     elf_flagscn     _((Elf_Scn *, Elf_Cmd, unsigned));
180 unsigned     elf_flagshdr    _((Elf_Scn *, Elf_Cmd, unsigned));
181 size_t       elf32_fsize     _((Elf_Type, size_t, unsigned));
182 Elf_Arhdr    *elf_getarhdr   _((Elf *));
183 Elf_Arsym    *elf_getarsym   _((Elf *, size_t *));
184 off_t        elf_getbase     _((Elf *));
185 Elf_Data     *elf_getdata    _((Elf_Scn *, Elf_Data *));
186 Elf32_Ehdr   *elf32_getehdr  _((Elf *));
187 char         *elf_getident   _((Elf *, size_t *));
188 Elf32_Phdr   *elf32_getphdr  _((Elf *));
189 Elf_Scn      *elf_getscn     _((Elf *, Elf *, size_t));
190 Elf32_Shdr   *elf32_getshdr  _((Elf_Scn *));
191 int          elf_getphnum    _((Elf *, size_t *));
192 int          elf_getphdrnum  _((Elf *, size_t *));
193 int          elf_getshnum    _((Elf *, size_t *));
194 int          elf_getshdrnum  _((Elf *, size_t *));
195 int          elf_getshstrndx _((Elf *, size_t *));
196 int          elf_getshdrstrndx _((Elf *, size_t *));
197 unsigned long elf_hash       _((const char *));
198 uint_t       elf_sys_encoding _((void));
199 long         elf32_checksum  _((Elf *));
200 Elf_Kind     elf_kind        _((Elf *));
201 Elf          *elf_memory     _((char *, size_t));
202 size_t       elf_ndxscn     _((Elf_Scn *));
203 Elf_Data     *elf_newdata    _((Elf_Scn *));
204 Elf32_Ehdr   *elf32_newehdr  _((Elf *));
205 Elf32_Phdr   *elf32_newphdr  _((Elf *, size_t));
206 Elf_Scn      *elf_newscn     _((Elf *));
207 Elf_Scn      *elf_nextscn   _((Elf *, Elf_Scn *));
208 Elf_Cmd      elf_next        _((Elf *));
209 size_t       elf_rand        _((Elf *, size_t));
210 Elf_Data     *elf_rawdata    _((Elf_Scn *, Elf_Data *));
211 char         *elf_rawfile    _((Elf *, size_t *));
212 char         *elf_strptr     _((Elf *, size_t, size_t));
213 off_t        elf_update      _((Elf *, Elf_Cmd));
214 unsigned     elf_version     _((unsigned));
215 Elf_Data     *elf32_xlatetof _((Elf_Data *, const Elf_Data *, unsigned));
216 Elf_Data     *elf32_xlatetom _((Elf_Data *, const Elf_Data *, unsigned));

```

```

208 #if defined(_LP64) || defined(_LONGLONG_TYPE)
209 size_t       elf64_fsize(Elf_Type, size_t, unsigned);
210 Elf64_Ehdr   *elf64_getehdr(Elf *);
211 Elf64_Phdr   *elf64_getphdr(Elf *);
212 Elf64_Shdr   *elf64_getshdr(Elf_Scn *);
213 long         elf64_checksum(Elf *);
214 Elf64_Ehdr   *elf64_newehdr(Elf *);
215 Elf64_Phdr   *elf64_newphdr(Elf *, size_t);
216 Elf_Data     *elf64_xlatetof(Elf_Data *, const Elf_Data *, unsigned);
217 Elf_Data     *elf64_xlatetom(Elf_Data *, const Elf_Data *, unsigned);
218 size_t       elf64_fsize _((Elf_Type, size_t, unsigned));
219 Elf64_Ehdr   *elf64_getehdr _((Elf *));
220 Elf64_Phdr   *elf64_getphdr _((Elf *));
221 Elf64_Shdr   *elf64_getshdr _((Elf_Scn *));
222 long         elf64_checksum _((Elf *));
223 Elf64_Ehdr   *elf64_newehdr _((Elf *));

```

```

225 Elf64_Phdr   *elf64_newphdr _((Elf *, size_t));
226 Elf_Data     *elf64_xlatetof _((Elf_Data *, const Elf_Data *, unsigned));
227 Elf_Data     *elf64_xlatetom _((Elf_Data *, const Elf_Data *, unsigned));
218 #endif /* (defined(_LP64) || defined(_LONGLONG_TYPE)) */

```

```

230 #undef _
220 #ifdef __cplusplus
221 }
_____unchanged_portion_omitted_

```