```
/* This file defines standard ELF types, structures, and macros.
   Copyright (C) 1995-2015 Free Software Foundation, Inc.
   This file is part of the GNU C Library.

   The GNU C Library is free software; you can redistribute it and/or
   modify it under the terms of the GNU Lesser General Public
   License as published by the Free Software Foundation; either
   version 2.1 of the License, or (at your option) any later version.

   The GNU C Library is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
   Lesser General Public License for more details.

   You should have received a copy of the GNU Lesser General Public
   License along with the GNU C Library; if not, see
   <http://www.gnu.org/licenses/>.  */

#ifndef _ELF_H
#define  _ELF_H 1

#include <features.h>

__BEGIN_DECLS

/* Standard ELF types.  */

#include <stdint.h>

/* Type for a 16-bit quantity.  */
typedef uint16_t Elf32_Half;
typedef uint16_t Elf64_Half;

/* Types for signed and unsigned 32-bit quantities.  */
typedef uint32_t Elf32_Word;
typedef  int32_t  Elf32_Sword;
typedef uint32_t Elf64_Word;
typedef  int32_t  Elf64_Sword;

/* Types for signed and unsigned 64-bit quantities.  */
typedef uint64_t Elf32_Xword;
typedef  int64_t  Elf32_Sxword;
typedef uint64_t Elf64_Xword;
typedef  int64_t  Elf64_Sxword;
```

```
/* Type of addresses.  */
typedef uint32_t Elf32_Addr;
typedef uint64_t Elf64_Addr;

/* Type of file offsets.  */
typedef uint32_t Elf32_Off;
typedef uint64_t Elf64_Off;

/* Type for section indices, which are 16-bit quantities.  */
typedef uint16_t Elf32_Section;
typedef uint16_t Elf64_Section;

/* Type for version symbol information.  */
typedef Elf32_Half Elf32_Versym;
typedef Elf64_Half Elf64_Versym;


/* The ELF file header.  This appears at the start of every ELF file.  */

#define EI_NIDENT (16)

typedef struct
{
  unsigned char  e_ident[EI_NIDENT];  /* Magic number and other info */
  Elf32_Half  e_type;           /* Object file type */
  Elf32_Half  e_machine;        /* Architecture */
  Elf32_Word e_version;         /* Object file version */
  Elf32_Addr  e_entry;      /* Entry point virtual address */
  Elf32_Off  e_phoff;       /* Program header table file offset */
  Elf32_Off  e_shoff;       /* Section header table file offset */
  Elf32_Word e_flags;       /* Processor-specific flags */
  Elf32_Half  e_ehsize;     /* ELF header size in bytes */
  Elf32_Half  e_phentsize;      /* Program header table entry size */
  Elf32_Half  e_phnum;          /* Program header table entry count */
  Elf32_Half  e_shentsize;      /* Section header table entry size */
  Elf32_Half  e_shnum;          /* Section header table entry count */
  Elf32_Half  e_shstrndx;       /* Section header string table index */
} Elf32_Ehdr;

typedef struct
{
  unsigned char  e_ident[EI_NIDENT];  /* Magic number and other info */
  Elf64_Half  e_type;           /* Object file type */
```

```
    Elf64_Half    e_machine;          /* Architecture */
    Elf64_Word e_version;             /* Object file version */
    Elf64_Addr  e_entry;        /* Entry point virtual address */
    Elf64_Off     e_phoff;        /* Program header table file offset */
    Elf64_Off     e_shoff;        /* Section header table file offset */
    Elf64_Word e_flags;         /* Processor-specific flags */
    Elf64_Half    e_ehsize;       /* ELF header size in bytes */
    Elf64_Half    e_phentsize;          /* Program header table entry size */
    Elf64_Half    e_phnum;              /* Program header table entry count */
    Elf64_Half    e_shentsize;          /* Section header table entry size */
    Elf64_Half    e_shnum;              /* Section header table entry count */
    Elf64_Half    e_shstrndx;           /* Section header string table index */
} Elf64_Ehdr;

/* Fields in the e_ident array.    The EI_* macros are indices into the
     array.    The macros under each EI_* macro are the values the byte
     may have.    */

#define EI_MAG0         0         /* File identification byte 0 index */
#define ELFMAG0         0x7f      /* Magic number byte 0 */

#define EI_MAG1         1         /* File identification byte 1 index */
#define ELFMAG1         'E'       /* Magic number byte 1 */

#define EI_MAG2         2         /* File identification byte 2 index */
#define ELFMAG2         'L'       /* Magic number byte 2 */

#define EI_MAG3         3         /* File identification byte 3 index */
#define ELFMAG3         'F'       /* Magic number byte 3 */

/* Conglomeration of the identification bytes, for easy testing as a word.    */
#define    ELFMAG       "\177ELF"
#define    SELFMAG      4

#define EI_CLASS    4           /* File class byte index */
#define ELFCLASSNONE  0         /* Invalid class */
#define ELFCLASS32      1         /* 32-bit objects */
#define ELFCLASS64      2         /* 64-bit objects */
#define ELFCLASSNUM   3

#define EI_DATA         5         /* Data encoding byte index */
#define ELFDATANONE   0         /* Invalid data encoding */
#define ELFDATA2LSB     1         /* 2's complement, little endian */
#define ELFDATA2MSB   2         /* 2's complement, big endian */
```

```
#define ELFDATANUM    3

#define EI_VERSION      6           /* File version byte index */
                        /* Value must be EV_CURRENT */

#define EI_OSABI    7           /* OS ABI identification */
#define ELFOSABI_NONE           0     /* UNIX System V ABI */
#define ELFOSABI_SYSV       0     /* Alias.    */
#define ELFOSABI_HPUX       1     /* HP-UX */
#define ELFOSABI_NETBSD         2     /* NetBSD.    */
#define ELFOSABI_GNU        3     /* Object uses GNU ELF extensions.    */
#define ELFOSABI_LINUX          ELFOSABI_GNU /* Compatibility alias.    */
#define ELFOSABI_SOLARIS    6     /* Sun Solaris.    */
#define ELFOSABI_AIX        7     /* IBM AIX.    */
#define ELFOSABI_IRIX       8     /* SGI Irix.    */
#define ELFOSABI_FREEBSD    9     /* FreeBSD.    */
#define ELFOSABI_TRU64          10    /* Compaq TRU64 UNIX.    */
#define ELFOSABI_MODESTO 11    /* Novell Modesto.    */
#define ELFOSABI_OPENBSD  12    /* OpenBSD.    */
#define ELFOSABI_ARM_AEABI      64    /* ARM EABI */
#define ELFOSABI_ARM        97    /* ARM */
#define ELFOSABI_STANDALONE    255  /* Standalone (embedded) application */

#define EI_ABIVERSION  8           /* ABI version */

#define EI_PAD          9           /* Byte index of padding bytes */

/* Legal values for e_type (object file type).    */

#define ET_NONE         0           /* No file type */
#define ET_REL          1           /* Relocatable file */
#define ET_EXEC         2           /* Executable file */
#define ET_DYN          3           /* Shared object file */
#define ET_CORE         4           /* Core file */
#define    ET_NUM       5           /* Number of defined types */
#define ET_LOOS         0xfe00          /* OS-specific range start */
#define ET_HIOS         0xfeff          /* OS-specific range end */
#define ET_LOPROC       0xff00          /* Processor-specific range start */
#define ET_HIPROC 0xffff           /* Processor-specific range end */

/* Legal values for e_machine (architecture).    */

#define EM_NONE          0          /* No machine */
#define EM_M32           1          /* AT&T WE 32100 */
```

```
#define EM_SPARC   2         /* SUN SPARC */
#define EM_386             3         /* Intel 80386 */
#define EM_68K             4         /* Motorola m68k family */
#define EM_88K             5         /* Motorola m88k family */
#define EM_860             7         /* Intel 80860 */
#define EM_MIPS            8         /* MIPS R3000 big-endian */
#define EM_S370            9         /* IBM System/370 */
#define EM_MIPS_RS3_LE     10          /* MIPS R3000 little-endian */

#define EM_PARISC15        /* HPPA */
#define EM_VPP500      17        /* Fujitsu VPP500 */
#define EM_SPARC32PLUS     18          /* Sun's "v8plus" */
#define EM_960             19        /* Intel 80960 */
#define EM_PPC             20        /* PowerPC */
#define EM_PPC64 21        /* PowerPC 64-bit */
#define EM_S390            22        /* IBM S390 */

#define EM_V800            36        /* NEC V800 series */
#define EM_FR20            37        /* Fujitsu FR20 */
#define EM_RH32            38        /* TRW RH-32 */
#define EM_RCE             39        /* Motorola RCE */
#define EM_ARM             40        /* ARM */
#define EM_FAKE_ALPHA      41          /* Digital Alpha */
#define EM_SH              42        /* Hitachi SH */
#define EM_SPARCV9     43        /* SPARC v9 64-bit */
#define EM_TRICORE     44        /* Siemens Tricore */
#define EM_ARC             45        /* Argonaut RISC Core */
#define EM_H8_300      46        /* Hitachi H8/300 */
#define EM_H8_300H     47        /* Hitachi H8/300H */
#define EM_H8S             48        /* Hitachi H8S */
#define EM_H8_500      49        /* Hitachi H8/500 */
#define EM_IA_64  50       /* Intel Merced */
#define EM_MIPS_X      51        /* Stanford MIPS-X */
#define EM_COLDFIRE    52        /* Motorola Coldfire */
#define EM_68HC12      53        /* Motorola M68HC12 */
#define EM_MMA             54        /* Fujitsu MMA Multimedia Accelerator*/
#define EM_PCP             55        /* Siemens PCP */
#define EM_NCPU            56        /* Sony nCPU embeeded RISC */
#define EM_NDR1            57        /* Denso NDR1 microprocessor */
#define EM_STARCORE    58        /* Motorola Start*Core processor */
#define EM_ME16            59        /* Toyota ME16 processor */
#define EM_ST100  60       /* STMicroelectronic ST100 processor */
#define EM_TINYJ   61      /* Advanced Logic Corp. Tinyj emb.fam*/
#define EM_X86_64      62        /* AMD x86-64 architecture */
```

```
#define EM_PDSP         63          /* Sony DSP Processor */

#define EM_FX66         66          /* Siemens FX66 microcontroller */
#define EM_ST9PLUS      67          /* STMicroelectronics ST9+ 8/16 mc */
#define EM_ST7          68          /* STmicroelectronics ST7 8 bit mc */
#define EM_68HC16       69          /* Motorola MC68HC16 microcontroller */
#define EM_68HC11       70          /* Motorola MC68HC11 microcontroller */
#define EM_68HC08       71          /* Motorola MC68HC08 microcontroller */
#define EM_68HC05       72          /* Motorola MC68HC05 microcontroller */
#define EM_SVX          73          /* Silicon Graphics SVx */
#define EM_ST19         74          /* STMicroelectronics ST19 8 bit mc */
#define EM_VAX          75          /* Digital VAX */
#define EM_CRIS         76          /* Axis Communications 32-bit embedded processor */
#define EM_JAVELIN      77          /* Infineon Technologies 32-bit embedded processor */
#define EM_FIREPATH     78          /* Element 14 64-bit DSP Processor */
#define EM_ZSP          79          /* LSI Logic 16-bit DSP Processor */
#define EM_MMIX         80          /* Donald Knuth's educational 64-bit processor */
#define EM_HUANY        81          /* Harvard University machine-independent object files */
#define EM_PRISM 82         /* SiTera Prism */
#define EM_AVR          83          /* Atmel AVR 8-bit microcontroller */
#define EM_FR30         84          /* Fujitsu FR30 */
#define EM_D10V         85          /* Mitsubishi D10V */
#define EM_D30V         86          /* Mitsubishi D30V */
#define EM_V850         87          /* NEC v850 */
#define EM_M32R         88          /* Mitsubishi M32R */
#define EM_MN10300      89          /* Matsushita MN10300 */
#define EM_MN10200      90          /* Matsushita MN10200 */
#define EM_PJ       91          /* picoJava */
#define EM_OPENRISC     92          /* OpenRISC 32-bit embedded processor */
#define EM_ARC_A5       93          /* ARC Cores Tangent-A5 */
#define EM_XTENSA       94          /* Tensilica Xtensa Architecture */
#define EM_ALTERA_NIOS2 113             /* Altera Nios II */
#define EM_AARCH64      183         /* ARM AARCH64 */
#define EM_TILEPRO      188         /* Tilera TILEPro */
#define EM_MICROBLAZE       189         /* Xilinx MicroBlaze */
#define EM_TILEGX 191           /* Tilera TILE-Gx */
#define EM_NUM          192

/* If it is necessary to assign new unofficial EM_* values, please
     pick large random numbers (0x8523, 0xa7f2, etc.) to minimize the
     chances of collision with official or non-GNU unofficial values.    */

#define EM_ALPHA 0x9026
```

```
/* Legal values for e_version (version).    */

#define EV_NONE         0           /* Invalid ELF version */
#define EV_CURRENT      1           /* Current version */
#define EV_NUM          2

/* Section header.    */

typedef struct
{
   Elf32_Word sh_name;      /* Section name (string tbl index) */
   Elf32_Word sh_type;      /* Section type */
   Elf32_Word sh_flags;     /* Section flags */
   Elf32_Addr  sh_addr;     /* Section virtual addr at execution */
   Elf32_Off     sh_offset;          /* Section file offset */
   Elf32_Word sh_size;      /* Section size in bytes */
   Elf32_Word sh_link;      /* Link to another section */
   Elf32_Word sh_info;      /* Additional section information */
   Elf32_Word sh_addralign;         /* Section alignment */
   Elf32_Word sh_entsize;           /* Entry size if section holds table */
} Elf32_Shdr;

typedef struct
{
   Elf64_Word sh_name;      /* Section name (string tbl index) */
   Elf64_Word sh_type;      /* Section type */
   Elf64_Xword      sh_flags;       /* Section flags */
   Elf64_Addr  sh_addr;     /* Section virtual addr at execution */
   Elf64_Off     sh_offset;          /* Section file offset */
   Elf64_Xword      sh_size;        /* Section size in bytes */
   Elf64_Word sh_link;      /* Link to another section */
   Elf64_Word sh_info;      /* Additional section information */
   Elf64_Xword      sh_addralign;       /* Section alignment */
   Elf64_Xword      sh_entsize;         /* Entry size if section holds table */
} Elf64_Shdr;

/* Special section indices.    */

#define SHN_UNDEF       0           /* Undefined section */
#define SHN_LORESERVE       0xff00          /* Start of reserved indices */
#define SHN_LOPROC      0xff00          /* Start of processor-specific */
#define SHN_BEFORE      0xff00          /* Order section before all others
                                    (Solaris).    */
#define SHN_AFTER       0xff01          /* Order section after all others
```

```
                                (Solaris).    */
#define SHN_HIPROC      0xff1f          /* End of processor-specific */
#define SHN_LOOS 0xff20         /* Start of OS-specific */
#define SHN_HIOS  0xff3f            /* End of OS-specific */
#define SHN_ABS         0xfff1          /* Associated symbol is absolute */
#define SHN_COMMON 0xfff2           /* Associated symbol is common */
#define SHN_XINDEX      0xffff          /* Index is in extra table.    */
#define SHN_HIRESERVE0xffff             /* End of reserved indices */

/* Legal values for sh_type (section type).    */

#define SHT_NULL      0       /* Section header table entry unused */
#define SHT_PROGBITS      1       /* Program data */
#define SHT_SYMTAB        2       /* Symbol table */
#define SHT_STRTAB        3       /* String table */
#define SHT_RELA      4       /* Relocation entries with addends */
#define SHT_HASH      5       /* Symbol hash table */
#define SHT_DYNAMIC       6       /* Dynamic linking information */
#define SHT_NOTE      7       /* Notes */
#define SHT_NOBITS        8       /* Program space with no data (bss) */
#define SHT_REL           9       /* Relocation entries, no addends */
#define SHT_SHLIB     10          /* Reserved */
#define SHT_DYNSYM        11          /* Dynamic linker symbol table */
#define SHT_INIT_ARRAY          14          /* Array of constructors */
#define SHT_FINI_ARRAY          15          /* Array of destructors */
#define SHT_PREINIT_ARRAY 16        /* Array of pre-constructors */
#define SHT_GROUP         17            /* Section group */
#define SHT_SYMTAB_SHNDX    18          /* Extended section indeces */
#define    SHT_NUM              19          /* Number of defined types.    */
#define SHT_LOOS      0x60000000/* Start OS-specific.    */
#define SHT_GNU_ATTRIBUTES 0x6ffffff5     /* Object attributes.    */
#define SHT_GNU_HASH          0x6ffffff6    /* GNU-style hash table.    */
#define SHT_GNU_LIBLIST       0x6ffffff7    /* Prelink library list */
#define SHT_CHECKSUM    0x6ffffff8    /* Checksum for DSO content.    */
#define SHT_LOSUNW        0x6ffffffa    /* Sun-specific low bound.    */
#define SHT_SUNW_move         0x6ffffffa
#define SHT_SUNW_COMDAT       0x6ffffffb
#define SHT_SUNW_syminfo    0x6ffffffc
#define SHT_GNU_verdef        0x6ffffffd    /* Version definition section.    */
#define SHT_GNU_verneed       0x6ffffffe    /* Version needs section.    */
#define SHT_GNU_versym        0x6fffffff    /* Version symbol table.    */
#define SHT_HISUNW        0x6fffffff    /* Sun-specific high bound.    */
#define SHT_HIOS      0x6fffffff    /* End OS-specific type */
#define SHT_LOPROC        0x70000000/* Start of processor-specific */
```

```
#define SHT_HIPROC      0x7fffffff      /* End of processor-specific */
#define SHT_LOUSER      0x80000000/* Start of application-specific */
#define SHT_HIUSER      0x8fffffff      /* End of application-specific */

/* Legal values for sh_flags (section flags).    */

#define SHF_WRITE       (1 << 0)  /* Writable */
#define SHF_ALLOC       (1 << 1)  /* Occupies memory during execution */
#define SHF_EXECINSTR       (1 << 2)  /* Executable */
#define SHF_MERGE           (1 << 4)  /* Might be merged */
#define SHF_STRINGS         (1 << 5)  /* Contains nul-terminated strings */
#define SHF_INFO_LINK       (1 << 6)  /* `sh_info' contains SHT index */
#define SHF_LINK_ORDER          (1 << 7)  /* Preserve order after combining */
#define SHF_OS_NONCONFORMING (1 << 8)       /* Non-standard OS specific handling
                            required */
#define SHF_GROUP           (1 << 9)  /* Section is member of a group.    */
#define SHF_TLS             (1 << 10)       /* Section hold thread-local data.    */
#define SHF_MASKOS          0x0ff00000    /* OS-specific.    */
#define SHF_MASKPROC        0xf0000000   /* Processor-specific */
#define SHF_ORDERED         (1 << 30)       /* Special ordering requirement
                            (Solaris).    */
#define SHF_EXCLUDE         (1 << 31)       /* Section is excluded unless
                            referenced or allocated (Solaris).*/

/* Section group handling.    */
#define GRP_COMDAT    0x1         /* Mark group as COMDAT.    */

/* Symbol table entry.    */

typedef struct
{
   Elf32_Word st_name;        /* Symbol name (string tbl index) */
   Elf32_Addr  st_value;        /* Symbol value */
   Elf32_Word st_size;          /* Symbol size */
   unsigned char     st_info;          /* Symbol type and binding */
   unsigned char     st_other;         /* Symbol visibility */
   Elf32_Section     st_shndx;         /* Section index */
} Elf32_Sym;

typedef struct
{
   Elf64_Word st_name;        /* Symbol name (string tbl index) */
   unsigned char     st_info;          /* Symbol type and binding */
   unsigned char st_other;           /* Symbol visibility */
```

```
   Elf64_Section      st_shndx;       /* Section index */
   Elf64_Addr  st_value;        /* Symbol value */
   Elf64_Xword       st_size;          /* Symbol size */
} Elf64_Sym;

/* The syminfo section if available contains additional information about
     every dynamic symbol.    */

typedef struct
{
   Elf32_Half si_boundto;            /* Direct bindings, symbol bound to */
   Elf32_Half si_flags;              /* Per symbol flags */
} Elf32_Syminfo;

typedef struct
{
   Elf64_Half si_boundto;            /* Direct bindings, symbol bound to */
   Elf64_Half si_flags;              /* Per symbol flags */
} Elf64_Syminfo;

/* Possible values for si_boundto.    */
#define SYMINFO_BT_SELF          0xffff       /* Symbol bound to self */
#define SYMINFO_BT_PARENT      0xfffe       /* Symbol bound to parent */
#define SYMINFO_BT_LOWRESERVE     0xff00      /* Beginning of reserved entries */

/* Possible bitmasks for si_flags.    */
#define SYMINFO_FLG_DIRECT     0x0001    /* Direct bound symbol */
#define SYMINFO_FLG_PASSTHRU 0x0002    /* Pass-thru symbol for translator */
#define SYMINFO_FLG_COPY 0x0004    /* Symbol is a copy-reloc */
#define SYMINFO_FLG_LAZYLOAD 0x0008    /* Symbol bound to object to be lazy
                                  loaded */
/* Syminfo version values.    */
#define SYMINFO_NONE             0
#define SYMINFO_CURRENT          1
#define SYMINFO_NUM        2


/* How to extract and insert information held in the st_info field.    */

#define ELF32_ST_BIND(val)          (((unsigned char) (val)) >> 4)
#define ELF32_ST_TYPE(val)          ((val) & 0xf)
#define ELF32_ST_INFO(bind, type)       (((bind) << 4) + ((type) & 0xf))

/* Both Elf32_Sym and Elf64_Sym use the same one-byte st_info field.    */
```

```
#define ELF64_ST_BIND(val)		ELF32_ST_BIND (val)
#define ELF64_ST_TYPE(val)		ELF32_ST_TYPE (val)
#define ELF64_ST_INFO(bind, type)	ELF32_ST_INFO ((bind), (type))

/* Legal values for ST_BIND subfield of st_info (symbol binding).	*/

#define STB_LOCAL 0		/* Local symbol */
#define STB_GLOBAL	1		/* Global symbol */
#define STB_WEAK 2		/* Weak symbol */
#define	STB_NUM		3		/* Number of defined types.	*/
#define STB_LOOS  10		/* Start of OS-specific */
#define STB_GNU_UNIQUE	10		/* Unique symbol.	*/
#define STB_HIOS   12		/* End of OS-specific */
#define STB_LOPROC	13		/* Start of processor-specific */
#define STB_HIPROC	15		/* End of processor-specific */

/* Legal values for ST_TYPE subfield of st_info (symbol type).	*/

#define STT_NOTYPE	0		/* Symbol type is unspecified */
#define STT_OBJECT	1		/* Symbol is a data object */
#define STT_FUNC  2		/* Symbol is a code object */
#define STT_SECTION	3		/* Symbol associated with a section */
#define STT_FILE	4		/* Symbol's name is file name */
#define STT_COMMON  5		/* Symbol is a common data object */
#define STT_TLS		6		/* Symbol is thread-local data object*/
#define	STT_NUM		7		/* Number of defined types.	*/
#define STT_LOOS  10		/* Start of OS-specific */
#define STT_GNU_IFUNC	10		/* Symbol is indirect code object */
#define STT_HIOS   12		/* End of OS-specific */
#define STT_LOPROC	13		/* Start of processor-specific */
#define STT_HIPROC	15		/* End of processor-specific */


/* Symbol table indices are found in the hash buckets and chain table
   of a symbol hash table section.  This special index value indicates
   the end of a chain, meaning no further symbols are found in that bucket.	*/

#define STN_UNDEF	0		/* End of a chain.	*/


/* How to extract and insert information held in the st_other field.	*/

#define ELF32_ST_VISIBILITY(o)	((o) & 0x03)
```

```
/* For ELF64 the definitions are the same.  */
#define ELF64_ST_VISIBILITY(o)  ELF32_ST_VISIBILITY (o)

/* Symbol visibility specification encoded in the st_other field.  */
#define STV_DEFAULT  0          /* Default symbol visibility rules */
#define STV_INTERNAL  1         /* Processor specific hidden class */
#define STV_HIDDEN  2           /* Sym unavailable in other modules */
#define STV_PROTECTED  3        /* Not preemptible, not exported */


/* Relocation table entry without addend (in section of type SHT_REL).  */

typedef struct
{
  Elf32_Addr  r_offset;     /* Address */
  Elf32_Word r_info;        /* Relocation type and symbol index */
} Elf32_Rel;

/* I have seen two different definitions of the Elf64_Rel and
   Elf64_Rela structures, so we'll leave them out until Novell (or
   whoever) gets their act together.  */
/* The following, at least, is used on Sparc v9, MIPS, and Alpha.  */

typedef struct
{
  Elf64_Addr  r_offset;     /* Address */
  Elf64_Xword   r_info;     /* Relocation type and symbol index */
} Elf64_Rel;

/* Relocation table entry with addend (in section of type SHT_RELA).  */

typedef struct
{
  Elf32_Addr  r_offset;     /* Address */
  Elf32_Word r_info;        /* Relocation type and symbol index */
  Elf32_Sword   r_addend;       /* Addend */
} Elf32_Rela;

typedef struct
{
  Elf64_Addr  r_offset;     /* Address */
  Elf64_Xword   r_info;     /* Relocation type and symbol index */
  Elf64_Sxword  r_addend;       /* Addend */
} Elf64_Rela;
```

```
/* How to extract and insert information held in the r_info field.    */

#define ELF32_R_SYM(val)            ((val) >> 8)
#define ELF32_R_TYPE(val)           ((val) & 0xff)
#define ELF32_R_INFO(sym, type)         (((sym) << 8) + ((type) & 0xff))

#define ELF64_R_SYM(i)              ((i) >> 32)
#define ELF64_R_TYPE(i)                 ((i) & 0xffffffff)
#define ELF64_R_INFO(sym,type)          ((((Elf64_Xword) (sym)) << 32) + (type))

/* Program segment header.    */

typedef struct
{
   Elf32_Word p_type;               /* Segment type */
   Elf32_Off     p_offset;         /* Segment file offset */
   Elf32_Addr  p_vaddr;         /* Segment virtual address */
   Elf32_Addr  p_paddr;         /* Segment physical address */
   Elf32_Word p_filesz;         /* Segment size in file */
   Elf32_Word p_memsz;              /* Segment size in memory */
   Elf32_Word p_flags;          /* Segment flags */
   Elf32_Word p_align;          /* Segment alignment */
} Elf32_Phdr;

typedef struct
{
   Elf64_Word p_type;               /* Segment type */
   Elf64_Word p_flags;          /* Segment flags */
   Elf64_Off     p_offset;         /* Segment file offset */
   Elf64_Addr  p_vaddr;         /* Segment virtual address */
   Elf64_Addr  p_paddr;         /* Segment physical address */
   Elf64_Xword       p_filesz;          /* Segment size in file */
   Elf64_Xword       p_memsz;               /* Segment size in memory */
   Elf64_Xword       p_align;           /* Segment alignment */
} Elf64_Phdr;

/* Special value for e_phnum.    This indicates that the real number of
     program headers is too large to fit into e_phnum.    Instead the real
     value is in the field sh_info of section 0.    */

#define PN_XNUM         0xffff

/* Legal values for p_type (segment type).    */
```

```
#define    PT_NULL         0            /* Program header table entry unused */
#define PT_LOAD            1            /* Loadable program segment */
#define PT_DYNAMIC     2            /* Dynamic linking information */
#define PT_INTERP 3            /* Program interpreter */
#define PT_NOTE            4            /* Auxiliary information */
#define PT_SHLIB    5          /* Reserved */
#define PT_PHDR            6            /* Entry for header table itself */
#define PT_TLS             7            /* Thread-local storage segment */
#define    PT_NUM          8            /* Number of defined types */
#define PT_LOOS            0x60000000    /* Start of OS-specific */
#define PT_GNU_EH_FRAME 0x6474e550    /* GCC .eh_frame_hdr segment */
#define PT_GNU_STACK 0x6474e551    /* Indicates stack executability */
#define PT_GNU_RELRO 0x6474e552    /* Read-only after relocation */
#define PT_LOSUNW      0x6ffffffa
#define PT_SUNWBSS     0x6ffffffa /* Sun Specific segment */
#define PT_SUNWSTACK 0x6ffffffb /* Stack segment */
#define PT_HISUNW      0x6fffffff
#define PT_HIOS            0x6fffffff  /* End of OS-specific */
#define PT_LOPROC      0x70000000    /* Start of processor-specific */
#define PT_HIPROC 0x7fffffff  /* End of processor-specific */

/* Legal values for p_flags (segment flags).    */

#define PF_X           (1 << 0)    /* Segment is executable */
#define PF_W           (1 << 1)    /* Segment is writable */
#define PF_R           (1 << 2)    /* Segment is readable */
#define PF_MASKOS      0x0ff00000     /* OS-specific */
#define PF_MASKPROC  0xf0000000     /* Processor-specific */

/* Legal values for note segment descriptor types for core files. */

#define NT_PRSTATUS    1            /* Contains copy of prstatus struct */
#define NT_FPREGSET    2            /* Contains copy of fpregset struct */
#define NT_PRPSINFO    3            /* Contains copy of prpsinfo struct */
#define NT_PRXREG      4            /* Contains copy of prxregset struct */
#define NT_TASKSTRUCT       4            /* Contains copy of task structure */
#define NT_PLATFORM    5            /* String from sysinfo(SI_PLATFORM) */
#define NT_AUXV            6            /* Contains copy of auxv array */
#define NT_GWINDOWS 7            /* Contains copy of gwindows struct */
#define NT_ASRS            8            /* Contains copy of asrset struct */
#define NT_PSTATUS     10           /* Contains copy of pstatus struct */
#define NT_PSINFO 13           /* Contains copy of psinfo struct */
#define NT_PRCRED      14           /* Contains copy of prcred struct */
```

```
#define NT_UTSNAME    15          /* Contains copy of utsname struct */
#define NT_LWPSTATUS  16          /* Contains copy of lwpstatus struct */
#define NT_LWPSINFO   17          /* Contains copy of lwpinfo struct */
#define NT_PRFPXREG   20          /* Contains copy of fprxregset struct */
#define NT_SIGINFO    0x53494749  /* Contains copy of siginfo_t,
                                     size might increase */
#define NT_FILE       0x46494c45  /* Contains information about mapped
                                     files */
#define NT_PRXFPREG   0x46e62b7f  /* Contains copy of user_fxsr_struct */
#define NT_PPC_VMX    0x100       /* PowerPC Altivec/VMX registers */
#define NT_PPC_SPE    0x101       /* PowerPC SPE/EVR registers */
#define NT_PPC_VSX    0x102       /* PowerPC VSX registers */
#define NT_386_TLS    0x200       /* i386 TLS slots (struct user_desc) */
#define NT_386_IOPERM     0x201       /* x86 io permission bitmap (1=deny) */
#define NT_X86_XSTATE 0x202       /* x86 extended state using xsave */
#define NT_S390_HIGH_GPRS     0x300   /* s390 upper register halves */
#define NT_S390_TIMER     0x301       /* s390 timer register */
#define NT_S390_TODCMP    0x302       /* s390 TOD clock comparator register */
#define NT_S390_TODPREG   0x303       /* s390 TOD programmable register */
#define NT_S390_CTRS  0x304       /* s390 control registers */
#define NT_S390_PREFIX    0x305       /* s390 prefix register */
#define NT_S390_LAST_BREAK    0x306   /* s390 breaking event address */
#define NT_S390_SYSTEM_CALL   0x307   /* s390 system call restart data */
#define NT_S390_TDB   0x308       /* s390 transaction diagnostic block */
#define NT_ARM_VFP    0x400       /* ARM VFP/NEON registers */
#define NT_ARM_TLS    0x401       /* ARM TLS register */
#define NT_ARM_HW_BREAK       0x402       /* ARM hardware breakpoint registers */
#define NT_ARM_HW_WATCH       0x403       /* ARM hardware watchpoint registers */

/* Legal values for the note segment descriptor types for object files.  */

#define NT_VERSION    1           /* Contains a version string.  */


/* Dynamic section entry.  */

typedef struct
{
  Elf32_Sword       d_tag;              /* Dynamic entry type */
  union
    {
      Elf32_Word d_val;               /* Integer value */
      Elf32_Addr d_ptr;               /* Address value */
    } d_un;
```

```
} Elf32_Dyn;

typedef struct
{
  Elf64_Sxword    d_tag;                  /* Dynamic entry type */
  union
    {
      Elf64_Xword d_val;          /* Integer value */
      Elf64_Addr d_ptr;                   /* Address value */
    } d_un;
} Elf64_Dyn;

/* Legal values for d_tag (dynamic entry type).    */

#define DT_NULL         0           /* Marks end of dynamic section */
#define DT_NEEDED       1           /* Name of needed library */
#define DT_PLTRELSZ     2           /* Size in bytes of PLT relocs */
#define DT_PLTGOT3          /* Processor defined value */
#define DT_HASH         4           /* Address of symbol hash table */
#define DT_STRTAB 5         /* Address of string table */
#define DT_SYMTAB       6           /* Address of symbol table */
#define DT_RELA         7           /* Address of Rela relocs */
#define DT_RELASZ 8         /* Total size of Rela relocs */
#define DT_RELAENT      9           /* Size of one Rela reloc */
#define DT_STRSZ   10       /* Size of string table */
#define DT_SYMENT       11          /* Size of one symbol table entry */
#define DT_INIT         12          /* Address of init function */
#define DT_FINI         13          /* Address of termination function */
#define DT_SONAME       14          /* Name of shared object */
#define DT_RPATH  15        /* Library search path (deprecated) */
#define DT_SYMBOLIC     16          /* Start symbol search here */
#define DT_REL          17          /* Address of Rel relocs */
#define DT_RELSZ   18       /* Total size of Rel relocs */
#define DT_RELENT19         /* Size of one Rel reloc */
#define DT_PLTREL 20        /* Type of reloc in PLT */
#define DT_DEBUG 21         /* For debugging; unspecified */
#define DT_TEXTREL      22          /* Reloc might modify .text */
#define DT_JMPREL       23          /* Address of PLT relocs */
#define    DT_BIND_NOW      24          /* Process relocations of object */
#define    DT_INIT_ARRAY    25          /* Array with addresses of init fct */
#define    DT_FINI_ARRAY    26          /* Array with addresses of fini fct */
#define    DT_INIT_ARRAYSZ  27          /* Size in bytes of DT_INIT_ARRAY */
#define    DT_FINI_ARRAYSZ  28          /* Size in bytes of DT_FINI_ARRAY */
#define DT_RUNPATH      29          /* Library search path */
```

```
#define DT_FLAGS  30          /* Flags for the object being loaded */
#define DT_ENCODING   32          /* Start of encoded range */
#define DT_PREINIT_ARRAY 32           /* Array with addresses of preinit fct*/
#define DT_PREINIT_ARRAYSZ 33         /* size in bytes of DT_PREINIT_ARRAY */
#define    DT_NUM         34          /* Number used */
#define DT_LOOS          0x6000000d    /* Start of OS-specific */
#define DT_HIOS          0x6ffff000       /* End of OS-specific */
#define DT_LOPROC        0x70000000    /* Start of processor-specific */
#define DT_HIPROC        0x7fffffff  /* End of processor-specific */
#define    DT_PROCNUMDT_MIPS_NUM       /* Most used by any processor */

/* DT_* entries which fall between DT_VALRNGHI & DT_VALRNGLO use the
     Dyn.d_un.d_val field of the Elf*_Dyn structure.    This follows Sun's
     approach.    */
#define DT_VALRNGLO   0x6ffffd00
#define DT_GNU_PRELINKED 0x6ffffdf5/* Prelinking timestamp */
#define DT_GNU_CONFLICTSZ 0x6ffffdf6      /* Size of conflict section */
#define DT_GNU_LIBLISTSZ 0x6ffffdf7   /* Size of library list */
#define DT_CHECKSUM  0x6ffffdf8
#define DT_PLTPADSZ     0x6ffffdf9
#define DT_MOVEENT    0x6ffffdfa
#define DT_MOVESZ       0x6ffffdfb
#define DT_FEATURE_1  0x6ffffdfc/* Feature selection (DTF_*).    */
#define DT_POSFLAG_1  0x6ffffdfd/* Flags for DT_* entries, effecting
                                   the following DT_* entry.    */
#define DT_SYMINSZ      0x6ffffdfe/* Size of syminfo table (in bytes) */
#define DT_SYMINENT   0x6ffffdff /* Entry size of syminfo */
#define DT_VALRNGHI   0x6ffffdff
#define DT_VALTAGIDX(tag)   (DT_VALRNGHI - (tag))    /* Reverse order! */
#define DT_VALNUM 12

/* DT_* entries which fall between DT_ADDRRNGHI & DT_ADDRRNGLO use the
     Dyn.d_un.d_ptr field of the Elf*_Dyn structure.

     If any adjustment is made to the ELF object after it has been
     built these entries will need to be adjusted.    */
#define DT_ADDRRNGLO       0x6ffffe00
#define DT_GNU_HASH  0x6ffffef5/* GNU-style hash table.    */
#define DT_TLSDESC_PLT       0x6ffffef6
#define DT_TLSDESC_GOT     0x6ffffef7
#define DT_GNU_CONFLICT   0x6ffffef8/* Start of conflict section */
#define DT_GNU_LIBLIST       0x6ffffef9/* Library list */
#define DT_CONFIG       0x6ffffefa/* Configuration information.    */
#define DT_DEPAUDIT    0x6ffffefb/* Dependency auditing.    */
```

```
#define DT_AUDIT  0x6ffffefc /* Object auditing.    */
#define    DT_PLTPAD      0x6ffffefd /* PLT padding.    */
#define    DT_MOVETAB 0x6ffffefe /* Move table.    */
#define DT_SYMINFO     0x6ffffeff /* Syminfo table.    */
#define DT_ADDRRNGHI 0x6ffffeff
#define DT_ADDRTAGIDX(tag)       (DT_ADDRRNGHI - (tag)) /* Reverse order! */
#define DT_ADDRNUM 11

/* The versioning entry types.    The next are defined as part of the
    GNU extension.    */
#define DT_VERSYM       0x6ffffff0

#define DT_RELACOUNT 0x6ffffff9
#define DT_RELCOUNT   0x6ffffffa

/* These were chosen by Sun.    */
#define DT_FLAGS_1      0x6ffffffb /* State flags, see DF_1_* below.    */
#define    DT_VERDEF     0x6ffffffc  /* Address of version definition
                                table */
#define    DT_VERDEFNUM     0x6ffffffd /* Number of version definitions */
#define    DT_VERNEED  0x6ffffffe /* Address of table with needed
                                versions */
#define    DT_VERNEEDNUM  0x6fffffff  /* Number of needed versions */
#define DT_VERSIONTAGIDX(tag)   (DT_VERNEEDNUM - (tag))     /* Reverse order! */
#define DT_VERSIONTAGNUM 16

/* Sun added these machine-independent extensions in the "processor-specific"
    range.    Be compatible.    */
#define DT_AUXILIARY       0x7ffffffd           /* Shared object to load before self */
#define DT_FILTER             0x7fffffff           /* Shared object to get values from */
#define DT_EXTRATAGIDX(tag)       ((Elf32_Word)-((Elf32_Sword) (tag) <<1>>1)-1)
#define DT_EXTRANUM  3

/* Values of `d_un.d_val' in the DT_FLAGS entry.    */
#define DF_ORIGIN 0x00000001    /* Object may use DF_ORIGIN */
#define DF_SYMBOLIC    0x00000002    /* Symbol resolutions starts here */
#define DF_TEXTREL      0x00000004    /* Object contains text relocations */
#define DF_BIND_NOW  0x00000008    /* No lazy binding for this object */
#define DF_STATIC_TLS   0x00000010    /* Module uses the static TLS model */

/* State flags selectable in the `d_un.d_val' element of the DT_FLAGS_1
    entry in the dynamic section.    */
#define DF_1_NOW        0x00000001    /* Set RTLD_NOW for this object.    */
#define DF_1_GLOBAL    0x00000002    /* Set RTLD_GLOBAL for this object.    */
```

```
#define DF_1_GROUP     0x00000004    /* Set RTLD_GROUP for this object.    */
#define DF_1_NODELETE       0x00000008    /* Set RTLD_NODELETE for this object.*/
#define DF_1_LOADFLTR0x00000010    /* Trigger filtee loading at runtime.*/
#define DF_1_INITFIRST 0x00000020    /* Set RTLD_INITFIRST for this object*/
#define DF_1_NOOPEN  0x00000040    /* Set RTLD_NOOPEN for this object.    */
#define DF_1_ORIGIN     0x00000080    /* $ORIGIN must be handled.    */
#define DF_1_DIRECT     0x00000100    /* Direct binding enabled.    */
#define DF_1_TRANS      0x00000200
#define DF_1_INTERPOSE      0x00000400    /* Object is used to interpose.    */
#define DF_1_NODEFLIB0x00000800    /* Ignore default lib search path.    */
#define DF_1_NODUMP 0x00001000    /* Object can't be dldump'ed.    */
#define DF_1_CONFALT  0x00002000    /* Configuration alternative created.*/
#define DF_1_ENDFILTEE       0x00004000    /* Filtee terminates filters search. */
#define    DF_1_DISPRELDNE  0x00008000    /* Disp reloc applied at build time. */
#define    DF_1_DISPRELPND 0x00010000    /* Disp reloc applied at run-time.    */
#define    DF_1_NODIRECT     0x00020000    /* Object has no-direct binding. */
#define    DF_1_IGNMULDEF  0x00040000
#define    DF_1_NOKSYMS     0x00080000
#define    DF_1_NOHDR  0x00100000
#define    DF_1_EDITED  0x00200000    /* Object is modified after built.    */
#define    DF_1_NORELOC      0x00400000
#define    DF_1_SYMINTPOSE 0x00800000    /* Object has individual interposers.    */
#define    DF_1_GLOBAUDIT   0x01000000    /* Global auditing required.    */
#define    DF_1_SINGLETON   0x02000000    /* Singleton symbols are used.    */

/* Flags for the feature selection in DT_FEATURE_1.    */
#define DTF_1_PARINIT  0x00000001
#define DTF_1_CONFEXP       0x00000002

/* Flags in the DT_POSFLAG_1 entry effecting only the next DT_* entry.    */
#define DF_P1_LAZYLOAD      0x00000001    /* Lazyload following object.    */
#define DF_P1_GROUPPERM 0x00000002    /* Symbols from next object are not
                                generally available.    */

/* Version definition sections.    */

typedef struct
{
   Elf32_Half    vd_version;           /* Version revision */
   Elf32_Half    vd_flags;        /* Version information */
   Elf32_Half    vd_ndx;               /* Version Index */
   Elf32_Half    vd_cnt;               /* Number of associated aux entries */
   Elf32_Word vd_hash;        /* Version name hash value */
   Elf32_Word vd_aux;               /* Offset in bytes to verdaux array */
```

```
   Elf32_Word vd_next;        /* Offset in bytes to next verdef
                              entry */
} Elf32_Verdef;

typedef struct
{
   Elf64_Half   vd_version;          /* Version revision */
   Elf64_Half   vd_flags;       /* Version information */
   Elf64_Half   vd_ndx;              /* Version Index */
   Elf64_Half   vd_cnt;              /* Number of associated aux entries */
   Elf64_Word vd_hash;       /* Version name hash value */
   Elf64_Word vd_aux;                /* Offset in bytes to verdaux array */
   Elf64_Word vd_next;       /* Offset in bytes to next verdef
                              entry */
} Elf64_Verdef;


/* Legal values for vd_version (version revision).    */
#define VER_DEF_NONE 0            /* No version */
#define VER_DEF_CURRENT   1             /* Current version */
#define VER_DEF_NUM  2            /* Given version number */

/* Legal values for vd_flags (version information flags).    */
#define VER_FLG_BASE   0x1        /* Version definition of file itself */
#define VER_FLG_WEAK 0x2          /* Weak version identifier */

/* Versym symbol index values.    */
#define    VER_NDX_LOCAL          0      /* Symbol is local.    */
#define    VER_NDX_GLOBAL         1      /* Symbol is global.    */
#define    VER_NDX_LORESERVE    0xff00      /* Beginning of reserved entries.    */
#define    VER_NDX_ELIMINATE    0xff01      /* Symbol is to be eliminated.    */

/* Auxialiary version information.    */

typedef struct
{
   Elf32_Word vda_name;              /* Version or dependency names */
   Elf32_Word vda_next;       /* Offset in bytes to next verdaux
                              entry */
} Elf32_Verdaux;

typedef struct
{
   Elf64_Word vda_name;              /* Version or dependency names */
```

```
  Elf64_Word vda_next;      /* Offset in bytes to next verdaux
                               entry */
} Elf64_Verdaux;


/* Version dependency section.    */

typedef struct
{
  Elf32_Half   vn_version;          /* Version of structure */
  Elf32_Half   vn_cnt;              /* Number of associated aux entries */
  Elf32_Word vn_file;         /* Offset of filename for this
                               dependency */
  Elf32_Word vn_aux;                /* Offset in bytes to vernaux array */
  Elf32_Word vn_next;       /* Offset in bytes to next verneed
                               entry */
} Elf32_Verneed;

typedef struct
{
  Elf64_Half   vn_version;          /* Version of structure */
  Elf64_Half   vn_cnt;              /* Number of associated aux entries */
  Elf64_Word vn_file;         /* Offset of filename for this
                               dependency */
  Elf64_Word vn_aux;                /* Offset in bytes to vernaux array */
  Elf64_Word vn_next;       /* Offset in bytes to next verneed
                               entry */
} Elf64_Verneed;


/* Legal values for vn_version (version revision).    */
#define VER_NEED_NONE       0           /* No version */
#define VER_NEED_CURRENT 1          /* Current version */
#define VER_NEED_NUM        2           /* Given version number */

/* Auxiliary needed version information.    */

typedef struct
{
  Elf32_Word vna_hash;              /* Hash value of dependency name */
  Elf32_Half   vna_flags;           /* Dependency specific information */
  Elf32_Half   vna_other;           /* Unused */
  Elf32_Word vna_name;              /* Dependency name string offset */
  Elf32_Word vna_next;              /* Offset in bytes to next vernaux
```

```
                                entry */
} Elf32_Vernaux;

typedef struct
{
   Elf64_Word vna_hash;             /* Hash value of dependency name */
   Elf64_Half    vna_flags;         /* Dependency specific information */
   Elf64_Half    vna_other;         /* Unused */
   Elf64_Word vna_name;             /* Dependency name string offset */
   Elf64_Word vna_next;             /* Offset in bytes to next vernaux
                                entry */
} Elf64_Vernaux;


/* Legal values for vna_flags.    */
#define VER_FLG_WEAK 0x2         /* Weak version identifier */


/* Auxiliary vector.    */

/* This vector is normally only used by the program interpreter.    The
     usual definition in an ABI supplement uses the name auxv_t.    The
     vector is not usually defined in a standard <elf.h> file, but it
     can't hurt.    We rename it to avoid conflicts.    The sizes of these
     types are an arrangement between the exec server and the program
     interpreter, so we don't fully specify them here.    */

typedef struct
{
   uint32_t a_type;        /* Entry type */
   union
      {
         uint32_t a_val;           /* Integer value */
         /* We use to have pointer elements added here.    We cannot do that,
        though, since it does not work when using 32-bit definitions
        on 64-bit platforms and vice versa.    */
      } a_un;
} Elf32_auxv_t;

typedef struct
{
   uint64_t a_type;        /* Entry type */
   union
      {
```

```
          uint64_t a_val;           /* Integer value */
          /* We use to have pointer elements added here.  We cannot do that,
         though, since it does not work when using 32-bit definitions
         on 64-bit platforms and vice versa.  */
       } a_un;
} Elf64_auxv_t;

#include <bits/auxv.h>
/* Note section contents.  Each entry in the note section begins with
     a header of a fixed form.  */

typedef struct
{
   Elf32_Word n_namesz;                 /* Length of the note's name.  */
   Elf32_Word n_descsz;             /* Length of the note's descriptor.  */
   Elf32_Word n_type;               /* Type of the note.  */
} Elf32_Nhdr;

typedef struct
{
   Elf64_Word n_namesz;                 /* Length of the note's name.  */
   Elf64_Word n_descsz;             /* Length of the note's descriptor.  */
   Elf64_Word n_type;               /* Type of the note.  */
} Elf64_Nhdr;

/* Known names of notes.  */

/* Solaris entries in the note section have this name.  */
#define ELF_NOTE_SOLARIS   "SUNW Solaris"

/* Note entries for GNU systems have this name.  */
#define ELF_NOTE_GNU        "GNU"


/* Defined types of notes for Solaris.  */

/* Value of descriptor (one word) is desired pagesize for the binary.  */
#define ELF_NOTE_PAGESIZE_HINT       1


/* Defined note types for GNU systems.  */

/* ABI information.  The descriptor consists of words:
     word 0: OS descriptor
```

```
     word 1: major version of the ABI
     word 2: minor version of the ABI
     word 3: subminor version of the ABI
*/
#define NT_GNU_ABI_TAG     1
#define ELF_NOTE_ABI   NT_GNU_ABI_TAG /* Old name.    */

/* Known OSes.    These values can appear in word 0 of an
     NT_GNU_ABI_TAG note section entry.    */
#define ELF_NOTE_OS_LINUX       0
#define ELF_NOTE_OS_GNU         1
#define ELF_NOTE_OS_SOLARIS2    2
#define ELF_NOTE_OS_FREEBSD     3

/* Synthetic hwcap information.    The descriptor begins with two words:
     word 0: number of entries
     word 1: bitmask of enabled entries
     Then follow variable-length entries, one byte followed by a
     '\0'-terminated hwcap name string.    The byte gives the bit
     number to test if enabled, (1U << bit) & bitmask.    */
#define NT_GNU_HWCAP      2

/* Build ID bits as generated by ld --build-id.
     The descriptor consists of any nonzero number of bytes.    */
#define NT_GNU_BUILD_ID    3

/* Version note generated by GNU gold containing a version string.    */
#define NT_GNU_GOLD_VERSION 4


/* Move records.    */
typedef struct
{
   Elf32_Xword m_value;             /* Symbol value.    */
   Elf32_Word m_info;          /* Size and index.    */
   Elf32_Word m_poffset;            /* Symbol offset.    */
   Elf32_Half m_repeat;        /* Repeat count.    */
   Elf32_Half m_stride;        /* Stride info.    */
} Elf32_Move;

typedef struct
{
   Elf64_Xword m_value;             /* Symbol value.    */
   Elf64_Xword m_info;         /* Size and index.    */
```

```
  Elf64_Xword m_poffset;    /* Symbol offset.    */
  Elf64_Half m_repeat;      /* Repeat count.    */
  Elf64_Half m_stride;      /* Stride info.    */
} Elf64_Move;

/* Macro to construct move records.    */
#define ELF32_M_SYM(info)  ((info) >> 8)
#define ELF32_M_SIZE(info)  ((unsigned char) (info))
#define ELF32_M_INFO(sym, size) (((sym) << 8) + (unsigned char) (size))

#define ELF64_M_SYM(info)  ELF32_M_SYM (info)
#define ELF64_M_SIZE(info)  ELF32_M_SIZE (info)
#define ELF64_M_INFO(sym, size) ELF32_M_INFO (sym, size)


/* Motorola 68k specific definitions.    */

/* Values for Elf32_Ehdr.e_flags.    */
#define EF_CPU32  0x00810000

/* m68k relocs.    */

#define R_68K_NONE    0       /* No reloc */
#define R_68K_32  1       /* Direct 32 bit    */
#define R_68K_16  2       /* Direct 16 bit    */
#define R_68K_8       3       /* Direct 8 bit    */
#define R_68K_PC32    4       /* PC relative 32 bit */
#define R_68K_PC16    5       /* PC relative 16 bit */
#define R_68K_PC8 6       /* PC relative 8 bit */
#define R_68K_GOT32   7       /* 32 bit PC relative GOT entry */
#define R_68K_GOT16   8       /* 16 bit PC relative GOT entry */
#define R_68K_GOT8    9       /* 8 bit PC relative GOT entry */
#define R_68K_GOT32O 10       /* 32 bit GOT offset */
#define R_68K_GOT16O 11       /* 16 bit GOT offset */
#define R_68K_GOT8O   12      /* 8 bit GOT offset */
#define R_68K_PLT32   13      /* 32 bit PC relative PLT address */
#define R_68K_PLT16   14      /* 16 bit PC relative PLT address */
#define R_68K_PLT8    15      /* 8 bit PC relative PLT address */
#define R_68K_PLT32O  16      /* 32 bit PLT offset */
#define R_68K_PLT16O  17      /* 16 bit PLT offset */
#define R_68K_PLT8O   18      /* 8 bit PLT offset */
#define R_68K_COPY    19      /* Copy symbol at runtime */
#define R_68K_GLOB_DAT    20      /* Create GOT entry */
#define R_68K_JMP_SLOT    21      /* Create PLT entry */
```

```
#define R_68K_RELATIVE      22          /* Adjust by program base */
#define R_68K_TLS_GD32          25              /* 32 bit GOT offset for GD */
#define R_68K_TLS_GD16          26              /* 16 bit GOT offset for GD */
#define R_68K_TLS_GD8           27              /* 8 bit GOT offset for GD */
#define R_68K_TLS_LDM32         28              /* 32 bit GOT offset for LDM */
#define R_68K_TLS_LDM16         29              /* 16 bit GOT offset for LDM */
#define R_68K_TLS_LDM8          30              /* 8 bit GOT offset for LDM */
#define R_68K_TLS_LDO32         31              /* 32 bit module-relative offset */
#define R_68K_TLS_LDO16         32              /* 16 bit module-relative offset */
#define R_68K_TLS_LDO8          33              /* 8 bit module-relative offset */
#define R_68K_TLS_IE32          34              /* 32 bit GOT offset for IE */
#define R_68K_TLS_IE16          35              /* 16 bit GOT offset for IE */
#define R_68K_TLS_IE8           36              /* 8 bit GOT offset for IE */
#define R_68K_TLS_LE32          37              /* 32 bit offset relative to
                            static TLS block */
#define R_68K_TLS_LE16          38              /* 16 bit offset relative to
                            static TLS block */
#define R_68K_TLS_LE8           39              /* 8 bit offset relative to
                            static TLS block */
#define R_68K_TLS_DTPMOD32  40              /* 32 bit module number */
#define R_68K_TLS_DTPREL32  41              /* 32 bit module-relative offset */
#define R_68K_TLS_TPREL32   42              /* 32 bit TP-relative offset */
/* Keep this the last entry.  */
#define R_68K_NUM   43

/* Intel 80386 specific definitions.  */

/* i386 relocs.  */

#define R_386_NONE      0       /* No reloc */
#define R_386_32    1       /* Direct 32 bit  */
#define R_386_PC32      2       /* PC relative 32 bit */
#define R_386_GOT32     3       /* 32 bit GOT entry */
#define R_386_PLT32     4       /* 32 bit PLT address */
#define R_386_COPY      5       /* Copy symbol at runtime */
#define R_386_GLOB_DAT      6       /* Create GOT entry */
#define R_386_JMP_SLOT      7       /* Create PLT entry */
#define R_386_RELATIVE      8       /* Adjust by program base */
#define R_386_GOTOFF    9       /* 32 bit offset to GOT */
#define R_386_GOTPC     10      /* 32 bit PC relative offset to GOT */
#define R_386_32PLT     11
#define R_386_TLS_TPOFF     14      /* Offset in static TLS block */
#define R_386_TLS_IE    15      /* Address of GOT entry for static TLS
                            block offset */
```

```
#define R_386_TLS_GOTIE         16          /* GOT entry for static TLS block
                                offset */
#define R_386_TLS_LE        17          /* Offset relative to static TLS
                                block */
#define R_386_TLS_GD        18          /* Direct 32 bit for GNU version of
                                general dynamic thread local data */
#define R_386_TLS_LDM           19          /* Direct 32 bit for GNU version of
                                local dynamic thread local data
                                in LE code */
#define R_386_16        20
#define R_386_PC16          21
#define R_386_8             22
#define R_386_PC8       23
#define R_386_TLS_GD_32         24          /* Direct 32 bit for general dynamic
                                thread local data */
#define R_386_TLS_GD_PUSH   25          /* Tag for pushl in GD TLS code */
#define R_386_TLS_GD_CALL   26      /* Relocation for call to
                                __tls_get_addr() */
#define R_386_TLS_GD_POP    27          /* Tag for popl in GD TLS code */
#define R_386_TLS_LDM_32    28          /* Direct 32 bit for local dynamic
                                thread local data in LE code */
#define R_386_TLS_LDM_PUSH 29           /* Tag for pushl in LDM TLS code */
#define R_386_TLS_LDM_CALL 30       /* Relocation for call to
                                __tls_get_addr() in LDM code */
#define R_386_TLS_LDM_POP   31          /* Tag for popl in LDM TLS code */
#define R_386_TLS_LDO_32    32      /* Offset relative to TLS block */
#define R_386_TLS_IE_32         33          /* GOT entry for negated static TLS
                                block offset */
#define R_386_TLS_LE_32         34          /* Negated offset relative to static
                                TLS block */
#define R_386_TLS_DTPMOD32 35           /* ID of module containing symbol */
#define R_386_TLS_DTPOFF32 36       /* Offset in TLS block */
#define R_386_TLS_TPOFF32   37      /* Negated offset in static TLS block */
#define R_386_SIZE32        38          /* 32-bit symbol size */
#define R_386_TLS_GOTDESC   39          /* GOT offset for TLS descriptor.   */
#define R_386_TLS_DESC_CALL 40      /* Marker of call through TLS
                                descriptor for
                                relaxation.   */
#define R_386_TLS_DESC      41      /* TLS descriptor containing
                                pointer to code and to
                                argument, returning the TLS
                                offset for the symbol.   */
#define R_386_IRELATIVE         42          /* Adjust indirectly by program base */
/* Keep this the last entry.  */
```

```
#define R_386_NUM           43

/* SUN SPARC specific definitions.    */

/* Legal values for ST_TYPE subfield of st_info (symbol type).    */

#define STT_SPARC_REGISTER      13    /* Global register reserved to app. */

/* Values for Elf64_Ehdr.e_flags.    */

#define EF_SPARCV9_MM           3
#define EF_SPARCV9_TSO          0
#define EF_SPARCV9_PSO          1
#define EF_SPARCV9_RMO          2
#define EF_SPARC_LEDATA         0x800000 /* little endian data */
#define EF_SPARC_EXT_MASK       0xFFFF00
#define EF_SPARC_32PLUS         0x000100 /* generic V8+ features */
#define EF_SPARC_SUN_US1 0x000200 /* Sun UltraSPARC1 extensions */
#define EF_SPARC_HAL_R1         0x000400 /* HAL R1 extensions */
#define EF_SPARC_SUN_US3 0x000800 /* Sun UltraSPARCIII extensions */

/* SPARC relocs.    */

#define R_SPARC_NONE        0     /* No reloc */
#define R_SPARC_8       1     /* Direct 8 bit */
#define R_SPARC_16          2     /* Direct 16 bit */
#define R_SPARC_32          3     /* Direct 32 bit */
#define R_SPARC_DISP8       4     /* PC relative 8 bit */
#define R_SPARC_DISP16          5     /* PC relative 16 bit */
#define R_SPARC_DISP32          6     /* PC relative 32 bit */
#define R_SPARC_WDISP30         7     /* PC relative 30 bit shifted */
#define R_SPARC_WDISP22         8     /* PC relative 22 bit shifted */
#define R_SPARC_HI22        9     /* High 22 bit */
#define R_SPARC_22          10    /* Direct 22 bit */
#define R_SPARC_13          11    /* Direct 13 bit */
#define R_SPARC_LO10        12    /* Truncated 10 bit */
#define R_SPARC_GOT10           13    /* Truncated 10 bit GOT entry */
#define R_SPARC_GOT13           14    /* 13 bit GOT entry */
#define R_SPARC_GOT22           15    /* 22 bit GOT entry shifted */
#define R_SPARC_PC10        16    /* PC relative 10 bit truncated */
#define R_SPARC_PC22        17    /* PC relative 22 bit shifted */
#define R_SPARC_WPLT30          18    /* 30 bit PC relative PLT address */
#define R_SPARC_COPY        19    /* Copy symbol at runtime */
#define R_SPARC_GLOB_DAT 20    /* Create GOT entry */
```

```
#define R_SPARC_JMP_SLOT  21   /* Create PLT entry */
#define R_SPARC_RELATIVE  22   /* Adjust by program base */
#define R_SPARC_UA32      23   /* Direct 32 bit unaligned */

/* Additional Sparc64 relocs.   */

#define R_SPARC_PLT32     24   /* Direct 32 bit ref to PLT entry */
#define R_SPARC_HIPLT22       25   /* High 22 bit PLT entry */
#define R_SPARC_LOPLT10       26   /* Truncated 10 bit PLT entry */
#define R_SPARC_PCPLT32       27   /* PC rel 32 bit ref to PLT entry */
#define R_SPARC_PCPLT22       28   /* PC rel high 22 bit PLT entry */
#define R_SPARC_PCPLT10       29   /* PC rel trunc 10 bit PLT entry */
#define R_SPARC_10        30   /* Direct 10 bit */
#define R_SPARC_11        31   /* Direct 11 bit */
#define R_SPARC_64        32   /* Direct 64 bit */
#define R_SPARC_OLO10         33   /* 10bit with secondary 13bit addend */
#define R_SPARC_HH22      34   /* Top 22 bits of direct 64 bit */
#define R_SPARC_HM10      35   /* High middle 10 bits of ... */
#define R_SPARC_LM22      36   /* Low middle 22 bits of ... */
#define R_SPARC_PC_HH22       37   /* Top 22 bits of pc rel 64 bit */
#define R_SPARC_PC_HM10       38   /* High middle 10 bit of ... */
#define R_SPARC_PC_LM22       39   /* Low miggle 22 bits of ... */
#define R_SPARC_WDISP16       40   /* PC relative 16 bit shifted */
#define R_SPARC_WDISP19       41   /* PC relative 19 bit shifted */
#define R_SPARC_GLOB_JMP 42   /* was part of v9 ABI but was removed */
#define R_SPARC_7     43   /* Direct 7 bit */
#define R_SPARC_5     44   /* Direct 5 bit */
#define R_SPARC_6     45   /* Direct 6 bit */
#define R_SPARC_DISP64        46   /* PC relative 64 bit */
#define R_SPARC_PLT64     47   /* Direct 64 bit ref to PLT entry */
#define R_SPARC_HIX22     48   /* High 22 bit complemented */
#define R_SPARC_LOX10     49   /* Truncated 11 bit complemented */
#define R_SPARC_H44       50   /* Direct high 12 of 44 bit */
#define R_SPARC_M44       51   /* Direct mid 22 of 44 bit */
#define R_SPARC_L44       52   /* Direct low 10 of 44 bit */
#define R_SPARC_REGISTER  53   /* Global register usage */
#define R_SPARC_UA64      54   /* Direct 64 bit unaligned */
#define R_SPARC_UA16      55   /* Direct 16 bit unaligned */
#define R_SPARC_TLS_GD_HI22   56
#define R_SPARC_TLS_GD_LO10   57
#define R_SPARC_TLS_GD_ADD    58
#define R_SPARC_TLS_GD_CALL   59
#define R_SPARC_TLS_LDM_HI22  60
#define R_SPARC_TLS_LDM_LO10 61
```

```
#define R_SPARC_TLS_LDM_ADD  62
#define R_SPARC_TLS_LDM_CALL  63
#define R_SPARC_TLS_LDO_HIX22 64
#define R_SPARC_TLS_LDO_LOX10        65
#define R_SPARC_TLS_LDO_ADD   66
#define R_SPARC_TLS_IE_HI22       67
#define R_SPARC_TLS_IE_LO10       68
#define R_SPARC_TLS_IE_LD  69
#define R_SPARC_TLS_IE_LDX        70
#define R_SPARC_TLS_IE_ADD        71
#define R_SPARC_TLS_LE_HIX22    72
#define R_SPARC_TLS_LE_LOX10   73
#define R_SPARC_TLS_DTPMOD32        74
#define R_SPARC_TLS_DTPMOD64        75
#define R_SPARC_TLS_DTPOFF32  76
#define R_SPARC_TLS_DTPOFF64  77
#define R_SPARC_TLS_TPOFF32     78
#define R_SPARC_TLS_TPOFF64     79
#define R_SPARC_GOTDATA_HIX22       80
#define R_SPARC_GOTDATA_LOX10       81
#define R_SPARC_GOTDATA_OP_HIX22 82
#define R_SPARC_GOTDATA_OP_LOX1083
#define R_SPARC_GOTDATA_OP     84
#define R_SPARC_H34           85
#define R_SPARC_SIZE32              86
#define R_SPARC_SIZE64              87
#define R_SPARC_WDISP10           88
#define R_SPARC_JMP_IREL   248
#define R_SPARC_IRELATIVE   249
#define R_SPARC_GNU_VTINHERIT       250
#define R_SPARC_GNU_VTENTRY  251
#define R_SPARC_REV32        252
/* Keep this the last entry.    */
#define R_SPARC_NUM          253

/* For Sparc64, legal values for d_tag of Elf64_Dyn.    */

#define DT_SPARC_REGISTER 0x70000001
#define DT_SPARC_NUM              2

/* MIPS R3000 specific definitions.    */

/* Legal values for e_flags field of Elf32_Ehdr.    */
```

```
#define EF_MIPS_NOREORDER      1         /* A .noreorder directive was used.    */
#define EF_MIPS_PIC            2         /* Contains PIC code.    */
#define EF_MIPS_CPIC           4         /* Uses PIC calling sequence.    */
#define EF_MIPS_XGOT           8
#define EF_MIPS_64BIT_WHIRL    16
#define EF_MIPS_ABI2           32
#define EF_MIPS_ABI_ON32       64
#define EF_MIPS_FP64           512    /* Uses FP64 (12 callee-saved).    */
#define EF_MIPS_NAN2008        1024    /* Uses IEEE 754-2008 NaN encoding.    */
#define EF_MIPS_ARCH           0xf0000000 /* MIPS architecture level.    */

/* Legal values for MIPS architecture level.    */

#define EF_MIPS_ARCH_1         0x00000000 /* -mips1 code.    */
#define EF_MIPS_ARCH_2         0x10000000 /* -mips2 code.    */
#define EF_MIPS_ARCH_3         0x20000000 /* -mips3 code.    */
#define EF_MIPS_ARCH_4         0x30000000 /* -mips4 code.    */
#define EF_MIPS_ARCH_5         0x40000000 /* -mips5 code.    */
#define EF_MIPS_ARCH_32        0x50000000 /* MIPS32 code.    */
#define EF_MIPS_ARCH_64        0x60000000 /* MIPS64 code.    */
#define EF_MIPS_ARCH_32R2      0x70000000 /* MIPS32r2 code.    */
#define EF_MIPS_ARCH_64R2      0x80000000 /* MIPS64r2 code.    */

/* The following are unofficial names and should not be used.    */

#define E_MIPS_ARCH_1          EF_MIPS_ARCH_1
#define E_MIPS_ARCH_2          EF_MIPS_ARCH_2
#define E_MIPS_ARCH_3          EF_MIPS_ARCH_3
#define E_MIPS_ARCH_4          EF_MIPS_ARCH_4
#define E_MIPS_ARCH_5          EF_MIPS_ARCH_5
#define E_MIPS_ARCH_32         EF_MIPS_ARCH_32
#define E_MIPS_ARCH_64         EF_MIPS_ARCH_64

/* Special section indices.    */

#define SHN_MIPS_ACOMMON    0xff00     /* Allocated common symbols.    */
#define SHN_MIPS_TEXT       0xff01     /* Allocated test symbols.    */
#define SHN_MIPS_DATA       0xff02     /* Allocated data symbols.    */
#define SHN_MIPS_SCOMMON    0xff03     /* Small common symbols.    */
#define SHN_MIPS_SUNDEFINED 0xff04     /* Small undefined symbols.    */

/* Legal values for sh_type field of Elf32_Shdr.    */

#define SHT_MIPS_LIBLIST    0x70000000 /* Shared objects used in link.    */
```

```
#define SHT_MIPS_MSYM           0x70000001
#define SHT_MIPS_CONFLICT0x70000002 /* Conflicting symbols.    */
#define SHT_MIPS_GPTAB          0x70000003 /* Global data area sizes.    */
#define SHT_MIPS_UCODE          0x70000004 /* Reserved for SGI/MIPS compilers */
#define SHT_MIPS_DEBUG          0x70000005 /* MIPS ECOFF debugging info.    */
#define SHT_MIPS_REGINFO  0x70000006 /* Register usage information.    */
#define SHT_MIPS_PACKAGE 0x70000007
#define SHT_MIPS_PACKSYM 0x70000008
#define SHT_MIPS_RELD           0x70000009
#define SHT_MIPS_IFACE          0x7000000b
#define SHT_MIPS_CONTENT0x7000000c
#define SHT_MIPS_OPTIONS  0x7000000d /* Miscellaneous options.    */
#define SHT_MIPS_SHDR           0x70000010
#define SHT_MIPS_FDESC          0x70000011
#define SHT_MIPS_EXTSYM         0x70000012
#define SHT_MIPS_DENSE          0x70000013
#define SHT_MIPS_PDESC          0x70000014
#define SHT_MIPS_LOCSYM         0x70000015
#define SHT_MIPS_AUXSYM         0x70000016
#define SHT_MIPS_OPTSYM         0x70000017
#define SHT_MIPS_LOCSTR         0x70000018
#define SHT_MIPS_LINE       0x70000019
#define SHT_MIPS_RFDESC         0x7000001a
#define SHT_MIPS_DELTASYM       0x7000001b
#define SHT_MIPS_DELTAINST      0x7000001c
#define SHT_MIPS_DELTACLASS     0x7000001d
#define SHT_MIPS_DWARF          0x7000001e /* DWARF debugging information.    */
#define SHT_MIPS_DELTADECL      0x7000001f
#define SHT_MIPS_SYMBOL_LIB   0x70000020
#define SHT_MIPS_EVENTS         0x70000021 /* Event section.    */
#define SHT_MIPS_TRANSLATE      0x70000022
#define SHT_MIPS_PIXIE      0x70000023
#define SHT_MIPS_XLATE          0x70000024
#define SHT_MIPS_XLATE_DEBUG 0x70000025
#define SHT_MIPS_WHIRL          0x70000026
#define SHT_MIPS_EH_REGION      0x70000027
#define SHT_MIPS_XLATE_OLD      0x70000028
#define SHT_MIPS_PDR_EXCEPTION      0x70000029

/* Legal values for sh_flags field of Elf32_Shdr.    */

#define SHF_MIPS_GPREL          0x10000000 /* Must be in global data area.    */
#define SHF_MIPS_MERGE          0x20000000
#define SHF_MIPS_ADDR           0x40000000
```

```
#define SHF_MIPS_STRINGS   0x80000000
#define SHF_MIPS_NOSTRIP   0x08000000
#define SHF_MIPS_LOCAL             0x04000000
#define SHF_MIPS_NAMES             0x02000000
#define SHF_MIPS_NODUPE            0x01000000


/* Symbol tables.    */

/* MIPS specific values for `st_other'.    */
#define STO_MIPS_DEFAULT           0x0
#define STO_MIPS_INTERNAL                0x1
#define STO_MIPS_HIDDEN                  0x2
#define STO_MIPS_PROTECTED               0x3
#define STO_MIPS_PLT                     0x8
#define STO_MIPS_SC_ALIGN_UNUSED0xff

/* MIPS specific values for `st_info'.    */
#define STB_MIPS_SPLIT_COMMON            13

/* Entries found in sections of type SHT_MIPS_GPTAB.    */

typedef union
{
   struct
     {
       Elf32_Word gt_current_g_value;       /* -G value used for compilation.    */
       Elf32_Word gt_unused;                /* Not used.    */
     } gt_header;                   /* First entry in section.    */
   struct
     {
       Elf32_Word gt_g_value;               /* If this value were used for -G.    */
       Elf32_Word gt_bytes;         /* This many bytes would be used.    */
     } gt_entry;                            /* Subsequent entries in section.    */
} Elf32_gptab;

/* Entry found in sections of type SHT_MIPS_REGINFO.    */

typedef struct
{
   Elf32_Word ri_gprmask;               /* General registers used.    */
   Elf32_Word ri_cprmask[4];            /* Coprocessor registers used.    */
   Elf32_Sword ri_gp_value;             /* $gp register value.    */
} Elf32_RegInfo;
```

```
/* Entries found in sections of type SHT_MIPS_OPTIONS.    */

typedef struct
{
   unsigned char kind;           /* Determines interpretation of the
                                 variable part of descriptor.    */
   unsigned char size;           /* Size of descriptor, including header.    */
   Elf32_Section section;/* Section header index of section affected,
                                 0 for global options.    */
   Elf32_Word info;       /* Kind-specific information.    */
} Elf_Options;

/* Values for `kind' field in Elf_Options.    */

#define ODK_NULL 0     /* Undefined.    */
#define ODK_REGINFO   1     /* Register usage information.    */
#define ODK_EXCEPTIONS     2     /* Exception processing options.    */
#define ODK_PAD          3     /* Section padding options.    */
#define ODK_HWPATCH 4     /* Hardware workarounds performed */
#define ODK_FILL    5     /* record the fill value used by the linker. */
#define ODK_TAGS 6     /* reserve space for desktop tools to write. */
#define ODK_HWAND    7     /* HW workarounds.    'AND' bits when merging. */
#define ODK_HWOR     8     /* HW workarounds.    'OR' bits when merging.    */

/* Values for `info' in Elf_Options for ODK_EXCEPTIONS entries.    */

#define OEX_FPU_MIN   0x1f /* FPE's which MUST be enabled.    */
#define OEX_FPU_MAX  0x1f00     /* FPE's which MAY be enabled.    */
#define OEX_PAGE0       0x10000  /* page zero must be mapped.    */
#define OEX_SMM         0x20000  /* Force sequential memory mode?    */
#define OEX_FPDBUG    0x40000  /* Force floating point debug mode?    */
#define OEX_PRECISEFP OEX_FPDBUG
#define OEX_DISMISS    0x80000  /* Dismiss invalid address faults?    */

#define OEX_FPU_INVAL0x10
#define OEX_FPU_DIV0  0x08
#define OEX_FPU_OFLO 0x04
#define OEX_FPU_UFLO 0x02
#define OEX_FPU_INEX  0x01

/* Masks for `info' in Elf_Options for an ODK_HWPATCH entry.    */

#define OHW_R4KEOP    0x1  /* R4000 end-of-page patch.    */
```

```
#define OHW_R8KPFETCH     0x2  /* may need R8000 prefetch patch.    */
#define OHW_R5KEOP    0x4  /* R5000 end-of-page patch.    */
#define OHW_R5KCVTL  0x8  /* R5000 cvt.[ds].l bug.    clean=1.    */

#define OPAD_PREFIX    0x1
#define OPAD_POSTFIX  0x2
#define OPAD_SYMBOL  0x4

/* Entry found in `.options' section.    */

typedef struct
{
   Elf32_Word hwp_flags1;    /* Extra flags.    */
   Elf32_Word hwp_flags2;    /* Extra flags.    */
} Elf_Options_Hw;

/* Masks for `info' in ElfOptions for ODK_HWAND and ODK_HWOR entries.    */

#define OHWA0_R4KEOP_CHECKED      0x00000001
#define OHWA1_R4KEOP_CLEAN   0x00000002

/* MIPS relocs.    */

#define R_MIPS_NONE         0     /* No reloc */
#define R_MIPS_16       1     /* Direct 16 bit */
#define R_MIPS_32       2     /* Direct 32 bit */
#define R_MIPS_REL32        3     /* PC relative 32 bit */
#define R_MIPS_26       4     /* Direct 26 bit shifted */
#define R_MIPS_HI16         5     /* High 16 bit */
#define R_MIPS_LO16         6     /* Low 16 bit */
#define R_MIPS_GPREL16          7     /* GP relative 16 bit */
#define R_MIPS_LITERAL          8     /* 16 bit literal entry */
#define R_MIPS_GOT16        9     /* 16 bit GOT entry */
#define R_MIPS_PC16         10    /* PC relative 16 bit */
#define R_MIPS_CALL16       11    /* 16 bit GOT entry for function */
#define R_MIPS_GPREL32          12    /* GP relative 32 bit */

#define R_MIPS_SHIFT5       16
#define R_MIPS_SHIFT6       17
#define R_MIPS_64       18
#define R_MIPS_GOT_DISP         19
#define R_MIPS_GOT_PAGE         20
#define R_MIPS_GOT_OFST         21
#define R_MIPS_GOT_HI16         22
```

```
#define R_MIPS_GOT_LO16         23
#define R_MIPS_SUB              24
#define R_MIPS_INSERT_A         25
#define R_MIPS_INSERT_B         26
#define R_MIPS_DELETE           27
#define R_MIPS_HIGHER           28
#define R_MIPS_HIGHEST          29
#define R_MIPS_CALL_HI16        30
#define R_MIPS_CALL_LO16        31
#define R_MIPS_SCN_DISP         32
#define R_MIPS_REL16            33
#define R_MIPS_ADD_IMMEDIATE    34
#define R_MIPS_PJUMP            35
#define R_MIPS_RELGOT           36
#define R_MIPS_JALR             37
#define R_MIPS_TLS_DTPMOD32     38  /* Module number 32 bit */
#define R_MIPS_TLS_DTPREL32     39  /* Module-relative offset 32 bit */
#define R_MIPS_TLS_DTPMOD64     40  /* Module number 64 bit */
#define R_MIPS_TLS_DTPREL64     41  /* Module-relative offset 64 bit */
#define R_MIPS_TLS_GD           42  /* 16 bit GOT offset for GD */
#define R_MIPS_TLS_LDM          43  /* 16 bit GOT offset for LDM */
#define R_MIPS_TLS_DTPREL_HI16  44  /* Module-relative offset, high 16 bits */
#define R_MIPS_TLS_DTPREL_LO16  45  /* Module-relative offset, low 16 bits */
#define R_MIPS_TLS_GOTTPREL     46  /* 16 bit GOT offset for IE */
#define R_MIPS_TLS_TPREL32      47  /* TP-relative offset, 32 bit */
#define R_MIPS_TLS_TPREL64      48  /* TP-relative offset, 64 bit */
#define R_MIPS_TLS_TPREL_HI16   49  /* TP-relative offset, high 16 bits */
#define R_MIPS_TLS_TPREL_LO16   50  /* TP-relative offset, low 16 bits */
#define R_MIPS_GLOB_DAT         51
#define R_MIPS_COPY             126
#define R_MIPS_JUMP_SLOT        127
/* Keep this the last entry.  */
#define R_MIPS_NUM              128

/* Legal values for p_type field of Elf32_Phdr.  */

#define PT_MIPS_REGINFO     0x70000000/* Register usage information. */
#define PT_MIPS_RTPROC      0x70000001/* Runtime procedure table. */
#define PT_MIPS_OPTIONS     0x70000002
#define PT_MIPS_ABIFLAGS    0x70000003  /* FP mode requirement. */

/* Special program header types.  */

#define PF_MIPS_LOCAL0x10000000
```

```
/* Legal values for d_tag field of Elf32_Dyn.  */

#define DT_MIPS_RLD_VERSION  0x70000001  /* Runtime linker interface version */
#define DT_MIPS_TIME_STAMP   0x70000002  /* Timestamp */
#define DT_MIPS_ICHECKSUM    0x70000003  /* Checksum */
#define DT_MIPS_IVERSION     0x70000004  /* Version string (string tbl index) */
#define DT_MIPS_FLAGS        0x70000005  /* Flags */
#define DT_MIPS_BASE_ADDRESS 0x70000006  /* Base address */
#define DT_MIPS_MSYM         0x70000007
#define DT_MIPS_CONFLICT     0x70000008  /* Address of CONFLICT section */
#define DT_MIPS_LIBLIST      0x70000009  /* Address of LIBLIST section */
#define DT_MIPS_LOCAL_GOTNO  0x7000000a  /* Number of local GOT entries */
#define DT_MIPS_CONFLICTNO   0x7000000b  /* Number of CONFLICT entries */
#define DT_MIPS_LIBLISTNO    0x70000010  /* Number of LIBLIST entries */
#define DT_MIPS_SYMTABNO     0x70000011  /* Number of DYNSYM entries */
#define DT_MIPS_UNREFEXTNO   0x70000012  /* First external DYNSYM */
#define DT_MIPS_GOTSYM       0x70000013  /* First GOT entry in DYNSYM */
#define DT_MIPS_HIPAGENO     0x70000014  /* Number of GOT page table entries */
#define DT_MIPS_RLD_MAP      0x70000016  /* Address of run time loader map.  */
#define DT_MIPS_DELTA_CLASS  0x70000017  /* Delta C++ class definition.  */
#define DT_MIPS_DELTA_CLASS_NO    0x70000018 /* Number of entries in
                                        DT_MIPS_DELTA_CLASS.  */
#define DT_MIPS_DELTA_INSTANCE    0x70000019 /* Delta C++ class instances.  */
#define DT_MIPS_DELTA_INSTANCE_NO 0x7000001a /* Number of entries in
                                        DT_MIPS_DELTA_INSTANCE.  */
#define DT_MIPS_DELTA_RELOC  0x7000001b /* Delta relocations.  */
#define DT_MIPS_DELTA_RELOC_NO 0x7000001c /* Number of entries in
                                        DT_MIPS_DELTA_RELOC.  */
#define DT_MIPS_DELTA_SYM    0x7000001d /* Delta symbols that Delta
                                       relocations refer to.  */
#define DT_MIPS_DELTA_SYM_NO 0x7000001e /* Number of entries in
                                       DT_MIPS_DELTA_SYM.  */
#define DT_MIPS_DELTA_CLASSSYM 0x70000020 /* Delta symbols that hold the
                                        class declaration.  */
#define DT_MIPS_DELTA_CLASSSYM_NO 0x70000021 /* Number of entries in
                                       DT_MIPS_DELTA_CLASSSYM.  */
#define DT_MIPS_CXX_FLAGS    0x70000022 /* Flags indicating for C++ flavor.  */
#define DT_MIPS_PIXIE_INIT   0x70000023
#define DT_MIPS_SYMBOL_LIB   0x70000024
#define DT_MIPS_LOCALPAGE_GOTIDX 0x70000025
#define DT_MIPS_LOCAL_GOTIDX 0x70000026
#define DT_MIPS_HIDDEN_GOTIDX 0x70000027
#define DT_MIPS_PROTECTED_GOTIDX 0x70000028
```

```
#define DT_MIPS_OPTIONS             0x70000029 /* Address of .options.    */
#define DT_MIPS_INTERFACE       0x7000002a /* Address of .interface.    */
#define DT_MIPS_DYNSTR_ALIGN 0x7000002b
#define DT_MIPS_INTERFACE_SIZE 0x7000002c /* Size of the .interface section. */
#define DT_MIPS_RLD_TEXT_RESOLVE_ADDR 0x7000002d /* Address of rld_text_rsolve
                                    function stored in GOT.    */
#define DT_MIPS_PERF_SUFFIX    0x7000002e /* Default suffix of dso to be added
                                by rld on dlopen() calls.    */
#define DT_MIPS_COMPACT_SIZE 0x7000002f /* (O32)Size of compact rel section. */
#define DT_MIPS_GP_VALUE          0x70000030 /* GP value for aux GOTs.    */
#define DT_MIPS_AUX_DYNAMIC    0x70000031 /* Address of aux .dynamic.    */
/* The address of .got.plt in an executable using the new non-PIC ABI.    */
#define DT_MIPS_PLTGOT              0x70000032
/* The base of the PLT in an executable using the new non-PIC ABI if that
     PLT is writable.    For a non-writable PLT, this is omitted or has a zero
     value.    */
#define DT_MIPS_RWPLT               0x70000034
#define DT_MIPS_NUM           0x35

/* Legal values for DT_MIPS_FLAGS Elf32_Dyn entry.    */

#define RHF_NONE              0               /* No flags */
#define RHF_QUICKSTART                (1 << 0)     /* Use quickstart */
#define RHF_NOTPOT                (1 << 1)     /* Hash size not power of 2 */
#define RHF_NO_LIBRARY_REPLACEMENT (1 << 2)      /* Ignore LD_LIBRARY_PATH */
#define RHF_NO_MOVE               (1 << 3)
#define RHF_SGI_ONLY              (1 << 4)
#define RHF_GUARANTEE_INIT          (1 << 5)
#define RHF_DELTA_C_PLUS_PLUS       (1 << 6)
#define RHF_GUARANTEE_START_INIT      (1 << 7)
#define RHF_PIXIE             (1 << 8)
#define RHF_DEFAULT_DELAY_LOAD          (1 << 9)
#define RHF_REQUICKSTART        (1 << 10)
#define RHF_REQUICKSTARTED          (1 << 11)
#define RHF_CORD              (1 << 12)
#define RHF_NO_UNRES_UNDEF        (1 << 13)
#define RHF_RLD_ORDER_SAFE          (1 << 14)

/* Entries found in sections of type SHT_MIPS_LIBLIST.    */

typedef struct
{
  Elf32_Word l_name;          /* Name (string table index) */
  Elf32_Word l_time_stamp; /* Timestamp */
```

```
  Elf32_Word l_checksum;    /* Checksum */
  Elf32_Word l_version;     /* Interface version */
  Elf32_Word l_flags;       /* Flags */
} Elf32_Lib;

typedef struct
{
  Elf64_Word l_name;        /* Name (string table index) */
  Elf64_Word l_time_stamp; /* Timestamp */
  Elf64_Word l_checksum;    /* Checksum */
  Elf64_Word l_version;     /* Interface version */
  Elf64_Word l_flags;       /* Flags */
} Elf64_Lib;


/* Legal values for l_flags.  */

#define LL_NONE             0
#define LL_EXACT_MATCH      (1 << 0)    /* Require exact match */
#define LL_IGNORE_INT_VER (1 << 1)  /* Ignore interface version */
#define LL_REQUIRE_MINOR    (1 << 2)
#define LL_EXPORTS          (1 << 3)
#define LL_DELAY_LOAD    (1 << 4)
#define LL_DELTA       (1 << 5)

/* Entries found in sections of type SHT_MIPS_CONFLICT.  */

typedef Elf32_Addr Elf32_Conflict;

typedef struct
{
  /* Version of flags structure.  */
  Elf32_Half version;
  /* The level of the ISA: 1-5, 32, 64.  */
  unsigned char isa_level;
  /* The revision of ISA: 0 for MIPS V and below, 1-n otherwise.  */
  unsigned char isa_rev;
  /* The size of general purpose registers.  */
  unsigned char gpr_size;
  /* The size of co-processor 1 registers.  */
  unsigned char cpr1_size;
  /* The size of co-processor 2 registers.  */
  unsigned char cpr2_size;
  /* The floating-point ABI.  */
```

```
  unsigned char fp_abi;
  /* Processor-specific extension.  */
  Elf32_Word isa_ext;
  /* Mask of ASEs used.  */
  Elf32_Word ases;
  /* Mask of general flags.  */
  Elf32_Word flags1;
  Elf32_Word flags2;
} Elf_MIPS_ABIFlags_v0;

/* Values for the register size bytes of an abi flags structure.  */

#define MIPS_AFL_REG_NONE      0x00  /* No registers.  */
#define MIPS_AFL_REG_32        0x01  /* 32-bit registers.  */
#define MIPS_AFL_REG_64        0x02  /* 64-bit registers.  */
#define MIPS_AFL_REG_128  0x03  /* 128-bit registers.  */

/* Masks for the ases word of an ABI flags structure.  */

#define MIPS_AFL_ASE_DSP  0x00000001 /* DSP ASE.  */
#define MIPS_AFL_ASE_DSPR2     0x00000002 /* DSP R2 ASE.  */
#define MIPS_AFL_ASE_EVA  0x00000004 /* Enhanced VA Scheme.  */
#define MIPS_AFL_ASE_MCU 0x00000008 /* MCU (MicroController) ASE.  */
#define MIPS_AFL_ASE_MDMX     0x00000010 /* MDMX ASE.  */
#define MIPS_AFL_ASE_MIPS3D    0x00000020 /* MIPS-3D ASE.  */
#define MIPS_AFL_ASE_MT        0x00000040 /* MT ASE.  */
#define MIPS_AFL_ASE_SMARTMIPS     0x00000080 /* SmartMIPS ASE.  */
#define MIPS_AFL_ASE_VIRT 0x00000100 /* VZ ASE.  */
#define MIPS_AFL_ASE_MSA 0x00000200 /* MSA ASE.  */
#define MIPS_AFL_ASE_MIPS16    0x00000400 /* MIPS16 ASE.  */
#define MIPS_AFL_ASE_MICROMIPS     0x00000800 /* MICROMIPS ASE.  */
#define MIPS_AFL_ASE_XPA  0x00001000 /* XPA ASE.  */
#define MIPS_AFL_ASE_MASK      0x00001fff /* All ASEs.  */

/* Values for the isa_ext word of an ABI flags structure.  */

#define MIPS_AFL_EXT_XLR       1      /* RMI Xlr instruction.  */
#define MIPS_AFL_EXT_OCTEON2     2      /* Cavium Networks Octeon2.  */
#define MIPS_AFL_EXT_OCTEONP     3      /* Cavium Networks OcteonP.  */
#define MIPS_AFL_EXT_LOONGSON_3A    4      /* Loongson 3A.  */
#define MIPS_AFL_EXT_OCTEON       5      /* Cavium Networks Octeon.  */
#define MIPS_AFL_EXT_5900     6      /* MIPS R5900 instruction.  */
#define MIPS_AFL_EXT_4650     7      /* MIPS R4650 instruction.  */
#define MIPS_AFL_EXT_4010     8      /* LSI R4010 instruction.  */
```

```
#define MIPS_AFL_EXT_4100     9      /* NEC VR4100 instruction.    */
#define MIPS_AFL_EXT_3900     10    /* Toshiba R3900 instruction.    */
#define MIPS_AFL_EXT_10000          11    /* MIPS R10000 instruction.    */
#define MIPS_AFL_EXT_SB1      12    /* Broadcom SB-1 instruction.    */
#define MIPS_AFL_EXT_4111     13    /* NEC VR4111/VR4181 instruction.    */
#define MIPS_AFL_EXT_4120     14    /* NEC VR4120 instruction.    */
#define MIPS_AFL_EXT_5400     15    /* NEC VR5400 instruction.    */
#define MIPS_AFL_EXT_5500     16    /* NEC VR5500 instruction.    */
#define MIPS_AFL_EXT_LOONGSON_2E    17    /* ST Microelectronics Loongson 2E.    */
#define MIPS_AFL_EXT_LOONGSON_2F    18    /* ST Microelectronics Loongson 2F.    */

/* Masks for the flags1 word of an ABI flags structure.    */
#define MIPS_AFL_FLAGS1_ODDSPREG    1    /* Uses odd single-precision registers.    */

/* Object attribute values.    */
enum
{
   /* Not tagged or not using any ABIs affected by the differences.    */
   Val_GNU_MIPS_ABI_FP_ANY = 0,
   /* Using hard-float -mdouble-float.    */
   Val_GNU_MIPS_ABI_FP_DOUBLE = 1,
   /* Using hard-float -msingle-float.    */
   Val_GNU_MIPS_ABI_FP_SINGLE = 2,
   /* Using soft-float.    */
   Val_GNU_MIPS_ABI_FP_SOFT = 3,
   /* Using -mips32r2 -mfp64.    */
   Val_GNU_MIPS_ABI_FP_OLD_64 = 4,
   /* Using -mfpxx.    */
   Val_GNU_MIPS_ABI_FP_XX = 5,
   /* Using -mips32r2 -mfp64.    */
   Val_GNU_MIPS_ABI_FP_64 = 6,
   /* Using -mips32r2 -mfp64 -mno-odd-spreg.    */
   Val_GNU_MIPS_ABI_FP_64A = 7,
   /* Maximum allocated FP ABI value.    */
   Val_GNU_MIPS_ABI_FP_MAX = 7
};

/* HPPA specific definitions.    */

/* Legal values for e_flags field of Elf32_Ehdr.    */

#define EF_PARISC_TRAPNIL  0x00010000 /* Trap nil pointer dereference.    */
#define EF_PARISC_EXT         0x00020000 /* Program uses arch. extensions. */
#define EF_PARISC_LSB         0x00040000 /* Program expects little endian. */
```

```
#define EF_PARISC_WIDE          0x00080000 /* Program expects wide mode.    */
#define EF_PARISC_NO_KABP       0x00100000 /* No kernel assisted branch
                                  prediction.    */
#define EF_PARISC_LAZYSWAP      0x00400000 /* Allow lazy swapping.    */
#define EF_PARISC_ARCH          0x0000ffff /* Architecture version.    */

/* Defined values for `e_flags & EF_PARISC_ARCH' are:    */

#define EFA_PARISC_1_0                  0x020b /* PA-RISC 1.0 big-endian.    */
#define EFA_PARISC_1_1                  0x0210 /* PA-RISC 1.1 big-endian.    */
#define EFA_PARISC_2_0                  0x0214 /* PA-RISC 2.0 big-endian.    */

/* Additional section indeces.    */

#define SHN_PARISC_ANSI_COMMON  0xff00          /* Section for tenatively declared
                                  symbols in ANSI C.    */
#define SHN_PARISC_HUGE_COMMON 0xff01           /* Common blocks in huge model.    */

/* Legal values for sh_type field of Elf32_Shdr.    */

#define SHT_PARISC_EXT          0x70000000 /* Contains product specific ext. */
#define SHT_PARISC_UNWIND       0x70000001 /* Unwind information.    */
#define SHT_PARISC_DOC          0x70000002 /* Debug info for optimized code. */

/* Legal values for sh_flags field of Elf32_Shdr.    */

#define SHF_PARISC_SHORT  0x20000000 /* Section with short addressing. */
#define SHF_PARISC_HUGE         0x40000000 /* Section far from gp.    */
#define SHF_PARISC_SBP          0x80000000 /* Static branch prediction code. */

/* Legal values for ST_TYPE subfield of st_info (symbol type).    */

#define STT_PARISC_MILLICODE    13    /* Millicode function entry point.    */

#define STT_HP_OPAQUE           (STT_LOOS + 0x1)
#define STT_HP_STUB         (STT_LOOS + 0x2)

/* HPPA relocs.    */

#define R_PARISC_NONE           0     /* No reloc.    */
#define R_PARISC_DIR32          1     /* Direct 32-bit reference.    */
#define R_PARISC_DIR21L         2     /* Left 21 bits of eff. address.    */
#define R_PARISC_DIR17R         3     /* Right 17 bits of eff. address.    */
#define R_PARISC_DIR17F         4     /* 17 bits of eff. address.    */
```

```
#define R_PARISC_DIR14R          6      /* Right 14 bits of eff. address.    */
#define R_PARISC_PCREL32    9      /* 32-bit rel. address.    */
#define R_PARISC_PCREL21L  10    /* Left 21 bits of rel. address.    */
#define R_PARISC_PCREL17R 11    /* Right 17 bits of rel. address.    */
#define R_PARISC_PCREL17F  12    /* 17 bits of rel. address.    */
#define R_PARISC_PCREL14R 14    /* Right 14 bits of rel. address.    */
#define R_PARISC_DPREL21L 18    /* Left 21 bits of rel. address.    */
#define R_PARISC_DPREL14R 22    /* Right 14 bits of rel. address.    */
#define R_PARISC_GPREL21L 26    /* GP-relative, left 21 bits.    */
#define R_PARISC_GPREL14R 30    /* GP-relative, right 14 bits.    */
#define R_PARISC_LTOFF21L  34    /* LT-relative, left 21 bits.    */
#define R_PARISC_LTOFF14R  38    /* LT-relative, right 14 bits.    */
#define R_PARISC_SECREL32  41    /* 32 bits section rel. address.    */
#define R_PARISC_SEGBASE   48    /* No relocation, set segment base.    */
#define R_PARISC_SEGREL32 49    /* 32 bits segment rel. address.    */
#define R_PARISC_PLTOFF21L50    /* PLT rel. address, left 21 bits.    */
#define R_PARISC_PLTOFF14R       54    /* PLT rel. address, right 14 bits.    */
#define R_PARISC_LTOFF_FPTR32  57    /* 32 bits LT-rel. function pointer. */
#define R_PARISC_LTOFF_FPTR21L58    /* LT-rel. fct ptr, left 21 bits. */
#define R_PARISC_LTOFF_FPTR14R       62    /* LT-rel. fct ptr, right 14 bits. */
#define R_PARISC_FPTR64          64    /* 64 bits function address.    */
#define R_PARISC_PLABEL32  65    /* 32 bits function address.    */
#define R_PARISC_PLABEL21L       66    /* Left 21 bits of fdesc address.    */
#define R_PARISC_PLABEL14R       70    /* Right 14 bits of fdesc address.    */
#define R_PARISC_PCREL64    72    /* 64 bits PC-rel. address.    */
#define R_PARISC_PCREL22F  74    /* 22 bits PC-rel. address.    */
#define R_PARISC_PCREL14WR     75    /* PC-rel. address, right 14 bits.    */
#define R_PARISC_PCREL14DR       76    /* PC rel. address, right 14 bits.    */
#define R_PARISC_PCREL16F  77    /* 16 bits PC-rel. address.    */
#define R_PARISC_PCREL16WF       78    /* 16 bits PC-rel. address.    */
#define R_PARISC_PCREL16DF       79    /* 16 bits PC-rel. address.    */
#define R_PARISC_DIR64           80    /* 64 bits of eff. address.    */
#define R_PARISC_DIR14WR  83    /* 14 bits of eff. address.    */
#define R_PARISC_DIR14DR   84    /* 14 bits of eff. address.    */
#define R_PARISC_DIR16F          85    /* 16 bits of eff. address.    */
#define R_PARISC_DIR16WF   86    /* 16 bits of eff. address.    */
#define R_PARISC_DIR16DF    87    /* 16 bits of eff. address.    */
#define R_PARISC_GPREL64    88    /* 64 bits of GP-rel. address.    */
#define R_PARISC_GPREL14WR     91    /* GP-rel. address, right 14 bits.    */
#define R_PARISC_GPREL14DR       92    /* GP-rel. address, right 14 bits.    */
#define R_PARISC_GPREL16F 93    /* 16 bits GP-rel. address.    */
#define R_PARISC_GPREL16WF       94    /* 16 bits GP-rel. address.    */
#define R_PARISC_GPREL16DF       95    /* 16 bits GP-rel. address.    */
#define R_PARISC_LTOFF64    96    /* 64 bits LT-rel. address.    */
```

```
#define R_PARISC_LTOFF14WR      99    /* LT-rel. address, right 14 bits.    */
#define R_PARISC_LTOFF14DR      100  /* LT-rel. address, right 14 bits.    */
#define R_PARISC_LTOFF16F  101  /* 16 bits LT-rel. address.    */
#define R_PARISC_LTOFF16WF      102  /* 16 bits LT-rel. address.    */
#define R_PARISC_LTOFF16DF      103  /* 16 bits LT-rel. address.    */
#define R_PARISC_SECREL64  104  /* 64 bits section rel. address.    */
#define R_PARISC_SEGREL64 112  /* 64 bits segment rel. address.    */
#define R_PARISC_PLTOFF14WR    115  /* PLT-rel. address, right 14 bits.    */
#define R_PARISC_PLTOFF14DR    116  /* PLT-rel. address, right 14 bits.    */
#define R_PARISC_PLTOFF16F      117  /* 16 bits LT-rel. address.    */
#define R_PARISC_PLTOFF16WF    118  /* 16 bits PLT-rel. address.    */
#define R_PARISC_PLTOFF16DF    119  /* 16 bits PLT-rel. address.    */
#define R_PARISC_LTOFF_FPTR64  120  /* 64 bits LT-rel. function ptr.    */
#define R_PARISC_LTOFF_FPTR14WR    123  /* LT-rel. fct. ptr., right 14 bits. */
#define R_PARISC_LTOFF_FPTR14DR    124  /* LT-rel. fct. ptr., right 14 bits. */
#define R_PARISC_LTOFF_FPTR16F      125  /* 16 bits LT-rel. function ptr.    */
#define R_PARISC_LTOFF_FPTR16WF    126  /* 16 bits LT-rel. function ptr.    */
#define R_PARISC_LTOFF_FPTR16DF    127  /* 16 bits LT-rel. function ptr.    */
#define R_PARISC_LORESERVE      128
#define R_PARISC_COPY       128  /* Copy relocation.    */
#define R_PARISC_IPLT       129  /* Dynamic reloc, imported PLT */
#define R_PARISC_EPLT       130  /* Dynamic reloc, exported PLT */
#define R_PARISC_TPREL32    153  /* 32 bits TP-rel. address.    */
#define R_PARISC_TPREL21L  154  /* TP-rel. address, left 21 bits.    */
#define R_PARISC_TPREL14R 158  /* TP-rel. address, right 14 bits.    */
#define R_PARISC_LTOFF_TP21L    162  /* LT-TP-rel. address, left 21 bits. */
#define R_PARISC_LTOFF_TP14R    166  /* LT-TP-rel. address, right 14 bits.*/
#define R_PARISC_LTOFF_TP14F    167  /* 14 bits LT-TP-rel. address.    */
#define R_PARISC_TPREL64    216  /* 64 bits TP-rel. address.    */
#define R_PARISC_TPREL14WR      219  /* TP-rel. address, right 14 bits.    */
#define R_PARISC_TPREL14DR      220  /* TP-rel. address, right 14 bits.    */
#define R_PARISC_TPREL16F  221  /* 16 bits TP-rel. address.    */
#define R_PARISC_TPREL16WF      222  /* 16 bits TP-rel. address.    */
#define R_PARISC_TPREL16DF      223  /* 16 bits TP-rel. address.    */
#define R_PARISC_LTOFF_TP64     224  /* 64 bits LT-TP-rel. address.    */
#define R_PARISC_LTOFF_TP14WR227  /* LT-TP-rel. address, right 14 bits.*/
#define R_PARISC_LTOFF_TP14DR 228  /* LT-TP-rel. address, right 14 bits.*/
#define R_PARISC_LTOFF_TP16F    229  /* 16 bits LT-TP-rel. address.    */
#define R_PARISC_LTOFF_TP16WF 230  /* 16 bits LT-TP-rel. address.    */
#define R_PARISC_LTOFF_TP16DF  231  /* 16 bits LT-TP-rel. address.    */
#define R_PARISC_GNU_VTENTRY 232
#define R_PARISC_GNU_VTINHERIT      233
#define R_PARISC_TLS_GD21L      234  /* GD 21-bit left.    */
#define R_PARISC_TLS_GD14R      235  /* GD 14-bit right.    */
```

```
#define R_PARISC_TLS_GDCALL     236  /* GD call to __t_g_a.    */
#define R_PARISC_TLS_LDM21L     237  /* LD module 21-bit left.    */
#define R_PARISC_TLS_LDM14R     238  /* LD module 14-bit right.    */
#define R_PARISC_TLS_LDMCALL    239  /* LD module call to __t_g_a.    */
#define R_PARISC_TLS_LDO21L     240  /* LD offset 21-bit left.    */
#define R_PARISC_TLS_LDO14R     241  /* LD offset 14-bit right.    */
#define R_PARISC_TLS_DTPMOD32       242  /* DTP module 32-bit.    */
#define R_PARISC_TLS_DTPMOD64       243  /* DTP module 64-bit.    */
#define R_PARISC_TLS_DTPOFF32 244  /* DTP offset 32-bit.    */
#define R_PARISC_TLS_DTPOFF64 245  /* DTP offset 32-bit.    */
#define R_PARISC_TLS_LE21L R_PARISC_TPREL21L
#define R_PARISC_TLS_LE14RR_PARISC_TPREL14R
#define R_PARISC_TLS_IE21L  R_PARISC_LTOFF_TP21L
#define R_PARISC_TLS_IE14R R_PARISC_LTOFF_TP14R
#define R_PARISC_TLS_TPREL32    R_PARISC_TPREL32
#define R_PARISC_TLS_TPREL64    R_PARISC_TPREL64
#define R_PARISC_HIRESERVE      255

/* Legal values for p_type field of Elf32_Phdr/Elf64_Phdr.    */

#define PT_HP_TLS       (PT_LOOS + 0x0)
#define PT_HP_CORE_NONE         (PT_LOOS + 0x1)
#define PT_HP_CORE_VERSION      (PT_LOOS + 0x2)
#define PT_HP_CORE_KERNEL       (PT_LOOS + 0x3)
#define PT_HP_CORE_COMM             (PT_LOOS + 0x4)
#define PT_HP_CORE_PROC         (PT_LOOS + 0x5)
#define PT_HP_CORE_LOADABLE   (PT_LOOS + 0x6)
#define PT_HP_CORE_STACK  (PT_LOOS + 0x7)
#define PT_HP_CORE_SHM          (PT_LOOS + 0x8)
#define PT_HP_CORE_MMF          (PT_LOOS + 0x9)
#define PT_HP_PARALLEL          (PT_LOOS + 0x10)
#define PT_HP_FASTBIND          (PT_LOOS + 0x11)
#define PT_HP_OPT_ANNOT         (PT_LOOS + 0x12)
#define PT_HP_HSL_ANNOT         (PT_LOOS + 0x13)
#define PT_HP_STACK           (PT_LOOS + 0x14)

#define PT_PARISC_ARCHEXT 0x70000000
#define PT_PARISC_UNWIND 0x70000001

/* Legal values for p_flags field of Elf32_Phdr/Elf64_Phdr.    */

#define PF_PARISC_SBP       0x08000000

#define PF_HP_PAGE_SIZE         0x00100000
```

```
#define PF_HP_FAR_SHARED 0x00200000
#define PF_HP_NEAR_SHARED      0x00400000
#define PF_HP_CODE             0x01000000
#define PF_HP_MODIFY           0x02000000
#define PF_HP_LAZYSWAP             0x04000000
#define PF_HP_SBP       0x08000000


/* Alpha specific definitions.    */

/* Legal values for e_flags field of Elf64_Ehdr.    */

#define EF_ALPHA_32BIT             1     /* All addresses must be < 2GB.    */
#define EF_ALPHA_CANRELAX      2     /* Relocations for relaxing exist.    */

/* Legal values for sh_type field of Elf64_Shdr.    */

/* These two are primerily concerned with ECOFF debugging info.    */
#define SHT_ALPHA_DEBUG        0x70000001
#define SHT_ALPHA_REGINFO      0x70000002

/* Legal values for sh_flags field of Elf64_Shdr.    */

#define SHF_ALPHA_GPREL            0x10000000

/* Legal values for st_other field of Elf64_Sym.    */
#define STO_ALPHA_NOPV             0x80/* No PV required.    */
#define STO_ALPHA_STD_GPLOAD       0x88/* PV only used for initial ldgp.    */

/* Alpha relocs.    */

#define R_ALPHA_NONE       0     /* No reloc */
#define R_ALPHA_REFLONG        1     /* Direct 32 bit */
#define R_ALPHA_REFQUAD        2     /* Direct 64 bit */
#define R_ALPHA_GPREL32        3     /* GP relative 32 bit */
#define R_ALPHA_LITERAL        4     /* GP relative 16 bit w/optimization */
#define R_ALPHA_LITUSE             5     /* Optimization hint for LITERAL */
#define R_ALPHA_GPDISP             6     /* Add displacement to GP */
#define R_ALPHA_BRADDR             7     /* PC+4 relative 23 bit shifted */
#define R_ALPHA_HINT       8     /* PC+4 relative 16 bit shifted */
#define R_ALPHA_SREL16             9     /* PC relative 16 bit */
#define R_ALPHA_SREL32             10    /* PC relative 32 bit */
#define R_ALPHA_SREL64             11    /* PC relative 64 bit */
#define R_ALPHA_GPRELHIGH      17    /* GP relative 32 bit, high 16 bits */
```

```
#define R_ALPHA_GPRELLOW       18    /* GP relative 32 bit, low 16 bits */
#define R_ALPHA_GPREL16        19    /* GP relative 16 bit */
#define R_ALPHA_COPY        24    /* Copy symbol at runtime */
#define R_ALPHA_GLOB_DAT 25    /* Create GOT entry */
#define R_ALPHA_JMP_SLOT 26    /* Create PLT entry */
#define R_ALPHA_RELATIVE   27    /* Adjust by program base */
#define R_ALPHA_TLS_GD_HI      28
#define R_ALPHA_TLSGD          29
#define R_ALPHA_TLS_LDM        30
#define R_ALPHA_DTPMOD64       31
#define R_ALPHA_GOTDTPREL      32
#define R_ALPHA_DTPREL64  33
#define R_ALPHA_DTPRELHI  34
#define R_ALPHA_DTPRELLO  35
#define R_ALPHA_DTPREL16  36
#define R_ALPHA_GOTTPREL 37
#define R_ALPHA_TPREL64        38
#define R_ALPHA_TPRELHI        39
#define R_ALPHA_TPRELLO        40
#define R_ALPHA_TPREL16        41
/* Keep this the last entry.    */
#define R_ALPHA_NUM        46

/* Magic values of the LITUSE relocation addend.    */
#define LITUSE_ALPHA_ADDR      0
#define LITUSE_ALPHA_BASE 1
#define LITUSE_ALPHA_BYTOFF    2
#define LITUSE_ALPHA_JSR    3
#define LITUSE_ALPHA_TLS_GD    4
#define LITUSE_ALPHA_TLS_LDM  5

/* Legal values for d_tag of Elf64_Dyn.    */
#define DT_ALPHA_PLTRO         (DT_LOPROC + 0)
#define DT_ALPHA_NUM           1

/* PowerPC specific declarations */

/* Values for Elf32/64_Ehdr.e_flags.    */
#define EF_PPC_EMB             0x80000000    /* PowerPC embedded flag */

/* Cygnus local bits below */
#define EF_PPC_RELOCATABLE      0x00010000    /* PowerPC -mrelocatable flag*/
#define EF_PPC_RELOCATABLE_LIB      0x00008000    /* PowerPC -mrelocatable-lib
                                   flag */
```

```
/* PowerPC relocations defined by the ABIs */
#define R_PPC_NONE          0
#define R_PPC_ADDR32        1     /* 32bit absolute address */
#define R_PPC_ADDR24        2     /* 26bit address, 2 bits ignored.    */
#define R_PPC_ADDR16        3     /* 16bit absolute address */
#define R_PPC_ADDR16_LO         4     /* lower 16bit of absolute address */
#define R_PPC_ADDR16_HI         5     /* high 16bit of absolute address */
#define R_PPC_ADDR16_HA         6     /* adjusted high 16bit */
#define R_PPC_ADDR14        7     /* 16bit address, 2 bits ignored */
#define R_PPC_ADDR14_BRTAKEN8
#define R_PPC_ADDR14_BRNTAKEN     9
#define R_PPC_REL24         10   /* PC relative 26 bit */
#define R_PPC_REL14         11   /* PC relative 16 bit */
#define R_PPC_REL14_BRTAKEN   12
#define R_PPC_REL14_BRNTAKEN 13
#define R_PPC_GOT16         14
#define R_PPC_GOT16_LO          15
#define R_PPC_GOT16_HI          16
#define R_PPC_GOT16_HA          17
#define R_PPC_PLTREL24          18
#define R_PPC_COPY          19
#define R_PPC_GLOB_DAT          20
#define R_PPC_JMP_SLOT          21
#define R_PPC_RELATIVE          22
#define R_PPC_LOCAL24PC         23
#define R_PPC_UADDR32           24
#define R_PPC_UADDR16           25
#define R_PPC_REL32         26
#define R_PPC_PLT32         27
#define R_PPC_PLTREL32          28
#define R_PPC_PLT16_LO          29
#define R_PPC_PLT16_HI          30
#define R_PPC_PLT16_HA          31
#define R_PPC_SDAREL16          32
#define R_PPC_SECTOFF       33
#define R_PPC_SECTOFF_LO  34
#define R_PPC_SECTOFF_HI  35
#define R_PPC_SECTOFF_HA  36

/* PowerPC relocations defined for the TLS access ABI.    */
#define R_PPC_TLS       67 /* none      (sym+add)@tls */
#define R_PPC_DTPMOD32          68 /* word32  (sym+add)@dtpmod */
#define R_PPC_TPREL16       69 /* half16*   (sym+add)@tprel */
```

```
#define R_PPC_TPREL16_LO  70 /* half16   (sym+add)@tprel@l */
#define R_PPC_TPREL16_HI  71 /* half16   (sym+add)@tprel@h */
#define R_PPC_TPREL16_HA  72 /* half16   (sym+add)@tprel@ha */
#define R_PPC_TPREL32     73 /* word32  (sym+add)@tprel */
#define R_PPC_DTPREL16        74 /* half16*  (sym+add)@dtprel */
#define R_PPC_DTPREL16_LO     75 /* half16   (sym+add)@dtprel@l */
#define R_PPC_DTPREL16_HI 76 /* half16   (sym+add)@dtprel@h */
#define R_PPC_DTPREL16_HA     77 /* half16   (sym+add)@dtprel@ha */
#define R_PPC_DTPREL32        78 /* word32  (sym+add)@dtprel */
#define R_PPC_GOT_TLSGD16     79 /* half16*  (sym+add)@got@tlsgd */
#define R_PPC_GOT_TLSGD16_LO 80 /* half16   (sym+add)@got@tlsgd@l */
#define R_PPC_GOT_TLSGD16_HI 81 /* half16   (sym+add)@got@tlsgd@h */
#define R_PPC_GOT_TLSGD16_HA82 /* half16   (sym+add)@got@tlsgd@ha */
#define R_PPC_GOT_TLSLD16     83 /* half16*  (sym+add)@got@tlsld */
#define R_PPC_GOT_TLSLD16_LO 84 /* half16   (sym+add)@got@tlsld@l */
#define R_PPC_GOT_TLSLD16_HI  85 /* half16   (sym+add)@got@tlsld@h */
#define R_PPC_GOT_TLSLD16_HA 86 /* half16   (sym+add)@got@tlsld@ha */
#define R_PPC_GOT_TPREL16     87 /* half16*  (sym+add)@got@tprel */
#define R_PPC_GOT_TPREL16_LO 88 /* half16   (sym+add)@got@tprel@l */
#define R_PPC_GOT_TPREL16_HI  89 /* half16   (sym+add)@got@tprel@h */
#define R_PPC_GOT_TPREL16_HA 90 /* half16   (sym+add)@got@tprel@ha */
#define R_PPC_GOT_DTPREL16    91 /* half16*  (sym+add)@got@dtprel */
#define R_PPC_GOT_DTPREL16_LO     92 /* half16*  (sym+add)@got@dtprel@l */
#define R_PPC_GOT_DTPREL16_HI     93 /* half16*  (sym+add)@got@dtprel@h */
#define R_PPC_GOT_DTPREL16_HA     94 /* half16*  (sym+add)@got@dtprel@ha */

/* The remaining relocs are from the Embedded ELF ABI, and are not
    in the SVR4 ELF ABI.    */
#define R_PPC_EMB_NADDR32    101
#define R_PPC_EMB_NADDR16    102
#define R_PPC_EMB_NADDR16_LO     103
#define R_PPC_EMB_NADDR16_HI     104
#define R_PPC_EMB_NADDR16_HA     105
#define R_PPC_EMB_SDAI16 106
#define R_PPC_EMB_SDA2I16     107
#define R_PPC_EMB_SDA2REL     108
#define R_PPC_EMB_SDA21       109  /* 16 bit offset in SDA */
#define R_PPC_EMB_MRKREF      110
#define R_PPC_EMB_RELSEC16    111
#define R_PPC_EMB_RELST_LO    112
#define R_PPC_EMB_RELST_HI    113
#define R_PPC_EMB_RELST_HA    114
#define R_PPC_EMB_BIT_FLD     115
#define R_PPC_EMB_RELSDA 116  /* 16 bit relative offset in SDA */
```

```
/* Diab tool relocations.  */
#define R_PPC_DIAB_SDA21_LO   180  /* like EMB_SDA21, but lower 16 bit */
#define R_PPC_DIAB_SDA21_HI   181  /* like EMB_SDA21, but high 16 bit */
#define R_PPC_DIAB_SDA21_HA   182  /* like EMB_SDA21, adjusted high 16 */
#define R_PPC_DIAB_RELSDA_LO  183  /* like EMB_RELSDA, but lower 16 bit */
#define R_PPC_DIAB_RELSDA_HI  184  /* like EMB_RELSDA, but high 16 bit */
#define R_PPC_DIAB_RELSDA_HA  185  /* like EMB_RELSDA, adjusted high 16 */

/* GNU extension to support local ifunc.  */
#define R_PPC_IRELATIVE           248

/* GNU relocs used in PIC code sequences.  */
#define R_PPC_REL16           249  /* half16     (sym+add-.) */
#define R_PPC_REL16_LO            250  /* half16     (sym+add-.)@l */
#define R_PPC_REL16_HI            251  /* half16     (sym+add-.)@h */
#define R_PPC_REL16_HA            252  /* half16     (sym+add-.)@ha */

/* This is a phony reloc to handle any old fashioned TOC16 references
   that may still be in object files.  */
#define R_PPC_TOC16           255

/* PowerPC specific values for the Dyn d_tag field.  */
#define DT_PPC_GOT            (DT_LOPROC + 0)
#define DT_PPC_NUM            1

/* PowerPC64 relocations defined by the ABIs */
#define R_PPC64_NONE        R_PPC_NONE
#define R_PPC64_ADDR32            R_PPC_ADDR32 /* 32bit absolute address */
#define R_PPC64_ADDR24            R_PPC_ADDR24 /* 26bit address, word aligned */
#define R_PPC64_ADDR16            R_PPC_ADDR16 /* 16bit absolute address */
#define R_PPC64_ADDR16_LO       R_PPC_ADDR16_LO       /* lower 16bits of address */
#define R_PPC64_ADDR16_HI       R_PPC_ADDR16_HI /* high 16bits of address. */
#define R_PPC64_ADDR16_HA       R_PPC_ADDR16_HA /* adjusted high 16bits.  */
#define R_PPC64_ADDR14            R_PPC_ADDR14 /* 16bit address, word aligned */
#define R_PPC64_ADDR14_BRTAKEN    R_PPC_ADDR14_BRTAKEN
#define R_PPC64_ADDR14_BRNTAKEN  R_PPC_ADDR14_BRNTAKEN
#define R_PPC64_REL24         R_PPC_REL24 /* PC-rel. 26 bit, word aligned */
#define R_PPC64_REL14         R_PPC_REL14 /* PC relative 16 bit */
#define R_PPC64_REL14_BRTAKEN       R_PPC_REL14_BRTAKEN
#define R_PPC64_REL14_BRNTAKEN     R_PPC_REL14_BRNTAKEN
#define R_PPC64_GOT16             R_PPC_GOT16
#define R_PPC64_GOT16_LO R_PPC_GOT16_LO
#define R_PPC64_GOT16_HI  R_PPC_GOT16_HI
```

```
#define R_PPC64_GOT16_HA R_PPC_GOT16_HA

#define R_PPC64_COPY        R_PPC_COPY
#define R_PPC64_GLOB_DAT R_PPC_GLOB_DAT
#define R_PPC64_JMP_SLOT  R_PPC_JMP_SLOT
#define R_PPC64_RELATIVE   R_PPC_RELATIVE

#define R_PPC64_UADDR32         R_PPC_UADDR32
#define R_PPC64_UADDR16         R_PPC_UADDR16
#define R_PPC64_REL32       R_PPC_REL32
#define R_PPC64_PLT32       R_PPC_PLT32
#define R_PPC64_PLTREL32   R_PPC_PLTREL32
#define R_PPC64_PLT16_LO   R_PPC_PLT16_LO
#define R_PPC64_PLT16_HI   R_PPC_PLT16_HI
#define R_PPC64_PLT16_HA  R_PPC_PLT16_HA

#define R_PPC64_SECTOFF          R_PPC_SECTOFF
#define R_PPC64_SECTOFF_LO     R_PPC_SECTOFF_LO
#define R_PPC64_SECTOFF_HI     R_PPC_SECTOFF_HI
#define R_PPC64_SECTOFF_HA     R_PPC_SECTOFF_HA
#define R_PPC64_ADDR30          37 /* word30 (S + A - P) >> 2 */
#define R_PPC64_ADDR64          38 /* doubleword64 S + A */
#define R_PPC64_ADDR16_HIGHER      39 /* half16 #higher(S + A) */
#define R_PPC64_ADDR16_HIGHERA    40 /* half16 #highera(S + A) */
#define R_PPC64_ADDR16_HIGHEST    41 /* half16 #highest(S + A) */
#define R_PPC64_ADDR16_HIGHESTA  42 /* half16 #highesta(S + A) */
#define R_PPC64_UADDR64         43 /* doubleword64 S + A */
#define R_PPC64_REL64       44 /* doubleword64 S + A - P */
#define R_PPC64_PLT64       45 /* doubleword64 L + A */
#define R_PPC64_PLTREL64   46 /* doubleword64 L + A - P */
#define R_PPC64_TOC16           47 /* half16* S + A - .TOC */
#define R_PPC64_TOC16_LO 48 /* half16 #lo(S + A - .TOC.) */
#define R_PPC64_TOC16_HI  49 /* half16 #hi(S + A - .TOC.) */
#define R_PPC64_TOC16_HA 50 /* half16 #ha(S + A - .TOC.) */
#define R_PPC64_TOC         51 /* doubleword64 .TOC */
#define R_PPC64_PLTGOT16  52 /* half16* M + A */
#define R_PPC64_PLTGOT16_LO   53 /* half16 #lo(M + A) */
#define R_PPC64_PLTGOT16_HI   54 /* half16 #hi(M + A) */
#define R_PPC64_PLTGOT16_HA   55 /* half16 #ha(M + A) */

#define R_PPC64_ADDR16_DS      56 /* half16ds* (S + A) >> 2 */
#define R_PPC64_ADDR16_LO_DS57 /* half16ds    #lo(S + A) >> 2 */
#define R_PPC64_GOT16_DS 58 /* half16ds* (G + A) >> 2 */
#define R_PPC64_GOT16_LO_DS   59 /* half16ds    #lo(G + A) >> 2 */
```

```
#define R_PPC64_PLT16_LO_DS    60 /* half16ds    #lo(L + A) >> 2 */
#define R_PPC64_SECTOFF_DS     61 /* half16ds* (R + A) >> 2 */
#define R_PPC64_SECTOFF_LO_DS      62 /* half16ds    #lo(R + A) >> 2 */
#define R_PPC64_TOC16_DS  63 /* half16ds* (S + A - .TOC.) >> 2 */
#define R_PPC64_TOC16_LO_DS   64 /* half16ds    #lo(S + A - .TOC.) >> 2 */
#define R_PPC64_PLTGOT16_DS    65 /* half16ds* (M + A) >> 2 */
#define R_PPC64_PLTGOT16_LO_DS     66 /* half16ds    #lo(M + A) >> 2 */

/* PowerPC64 relocations defined for the TLS access ABI.    */
#define R_PPC64_TLS            67 /* none      (sym+add)@tls */
#define R_PPC64_DTPMOD64       68 /* doubleword64 (sym+add)@dtpmod */
#define R_PPC64_TPREL16            69 /* half16*   (sym+add)@tprel */
#define R_PPC64_TPREL16_LO     70 /* half16    (sym+add)@tprel@l */
#define R_PPC64_TPREL16_HI     71 /* half16    (sym+add)@tprel@h */
#define R_PPC64_TPREL16_HA     72 /* half16    (sym+add)@tprel@ha */
#define R_PPC64_TPREL64            73 /* doubleword64 (sym+add)@tprel */
#define R_PPC64_DTPREL16  74 /* half16*   (sym+add)@dtprel */
#define R_PPC64_DTPREL16_LO    75 /* half16    (sym+add)@dtprel@l */
#define R_PPC64_DTPREL16_HI    76 /* half16    (sym+add)@dtprel@h */
#define R_PPC64_DTPREL16_HA    77 /* half16    (sym+add)@dtprel@ha */
#define R_PPC64_DTPREL64  78 /* doubleword64 (sym+add)@dtprel */
#define R_PPC64_GOT_TLSGD16    79 /* half16*   (sym+add)@got@tlsgd */
#define R_PPC64_GOT_TLSGD16_LO     80 /* half16    (sym+add)@got@tlsgd@l */
#define R_PPC64_GOT_TLSGD16_HI     81 /* half16    (sym+add)@got@tlsgd@h */
#define R_PPC64_GOT_TLSGD16_HA     82 /* half16    (sym+add)@got@tlsgd@ha */
#define R_PPC64_GOT_TLSLD16    83 /* half16*   (sym+add)@got@tlsld */
#define R_PPC64_GOT_TLSLD16_LO     84 /* half16    (sym+add)@got@tlsld@l */
#define R_PPC64_GOT_TLSLD16_HI     85 /* half16    (sym+add)@got@tlsld@h */
#define R_PPC64_GOT_TLSLD16_HA     86 /* half16    (sym+add)@got@tlsld@ha */
#define R_PPC64_GOT_TPREL16_DS     87 /* half16ds*       (sym+add)@got@tprel */
#define R_PPC64_GOT_TPREL16_LO_DS 88 /* half16ds (sym+add)@got@tprel@l */
#define R_PPC64_GOT_TPREL16_HI     89 /* half16    (sym+add)@got@tprel@h */
#define R_PPC64_GOT_TPREL16_HA     90 /* half16    (sym+add)@got@tprel@ha */
#define R_PPC64_GOT_DTPREL16_DS    91 /* half16ds*       (sym+add)@got@dtprel */
#define R_PPC64_GOT_DTPREL16_LO_DS 92 /* half16ds (sym+add)@got@dtprel@l */
#define R_PPC64_GOT_DTPREL16_HI    93 /* half16    (sym+add)@got@dtprel@h */
#define R_PPC64_GOT_DTPREL16_HA  94 /* half16    (sym+add)@got@dtprel@ha */
#define R_PPC64_TPREL16_DS     95 /* half16ds*       (sym+add)@tprel */
#define R_PPC64_TPREL16_LO_DS      96 /* half16ds (sym+add)@tprel@l */
#define R_PPC64_TPREL16_HIGHER     97 /* half16    (sym+add)@tprel@higher */
#define R_PPC64_TPREL16_HIGHERA    98 /* half16    (sym+add)@tprel@highera */
#define R_PPC64_TPREL16_HIGHEST    99 /* half16    (sym+add)@tprel@highest */
#define R_PPC64_TPREL16_HIGHESTA 100 /* half16    (sym+add)@tprel@highesta */
#define R_PPC64_DTPREL16_DS    101 /* half16ds* (sym+add)@dtprel */
```

```
#define R_PPC64_DTPREL16_LO_DS     102 /* half16ds       (sym+add)@dtprel@l */
#define R_PPC64_DTPREL16_HIGHER    103 /* half16   (sym+add)@dtprel@higher */
#define R_PPC64_DTPREL16_HIGHERA 104 /* half16   (sym+add)@dtprel@highera */
#define R_PPC64_DTPREL16_HIGHEST 105 /* half16    (sym+add)@dtprel@highest */
#define R_PPC64_DTPREL16_HIGHESTA 106 /* half16 (sym+add)@dtprel@highesta */
#define R_PPC64_TLSGD              107 /* none     (sym+add)@tlsgd */
#define R_PPC64_TLSLD        108 /* none     (sym+add)@tlsld */
#define R_PPC64_TOCSAVE            109 /* none */

/* Added when HA and HI relocs were changed to report overflows.    */
#define R_PPC64_ADDR16_HIGH   110
#define R_PPC64_ADDR16_HIGHA111
#define R_PPC64_TPREL16_HIGH  112
#define R_PPC64_TPREL16_HIGHA       113
#define R_PPC64_DTPREL16_HIGH       114
#define R_PPC64_DTPREL16_HIGHA      115

/* GNU extension to support local ifunc.    */
#define R_PPC64_JMP_IREL   247
#define R_PPC64_IRELATIVE   248
#define R_PPC64_REL16        249  /* half16      (sym+add-.) */
#define R_PPC64_REL16_LO  250  /* half16      (sym+add-.)@l */
#define R_PPC64_REL16_HI   251  /* half16      (sym+add-.)@h */
#define R_PPC64_REL16_HA  252  /* half16      (sym+add-.)@ha */

/* e_flags bits specifying ABI.
     1 for original function descriptor using ABI,
     2 for revised ABI without function descriptors,
     0 for unspecified or not using any features affected by the differences.    */
#define EF_PPC64_ABI    3

/* PowerPC64 specific values for the Dyn d_tag field.    */
#define DT_PPC64_GLINK    (DT_LOPROC + 0)
#define DT_PPC64_OPD (DT_LOPROC + 1)
#define DT_PPC64_OPDSZ     (DT_LOPROC + 2)
#define DT_PPC64_OPT  (DT_LOPROC + 3)
#define DT_PPC64_NUM        4

/* PowerPC64 specific values for the DT_PPC64_OPT Dyn entry.    */
#define PPC64_OPT_TLS        1
#define PPC64_OPT_MULTI_TOC   2

/* PowerPC64 specific values for the Elf64_Sym st_other field.    */
#define STO_PPC64_LOCAL_BIT     5
```

```
#define STO_PPC64_LOCAL_MASK(7 << STO_PPC64_LOCAL_BIT)
#define PPC64_LOCAL_ENTRY_OFFSET(other)                          \
  (((1 << (((other) & STO_PPC64_LOCAL_MASK) >> STO_PPC64_LOCAL_BIT)) >> 2) << 2)


/* ARM specific declarations */

/* Processor specific flags for the ELF header e_flags field.    */
#define EF_ARM_RELEXEC          0x01
#define EF_ARM_HASENTRY         0x02
#define EF_ARM_INTERWORK        0x04
#define EF_ARM_APCS_26          0x08
#define EF_ARM_APCS_FLOAT       0x10
#define EF_ARM_PIC          0x20
#define EF_ARM_ALIGN8           0x40 /* 8-bit structure alignment is in use */
#define EF_ARM_NEW_ABI          0x80
#define EF_ARM_OLD_ABI          0x100
#define EF_ARM_SOFT_FLOAT       0x200
#define EF_ARM_VFP_FLOAT 0x400
#define EF_ARM_MAVERICK_FLOAT       0x800

#define EF_ARM_ABI_FLOAT_SOFT       0x200       /* NB conflicts with EF_ARM_SOFT_FLOAT */
#define EF_ARM_ABI_FLOAT_HARD       0x400       /* NB conflicts with EF_ARM_VFP_FLOAT */


/* Other constants defined in the ARM ELF spec. version B-01.    */
/* NB. These conflict with values defined above.    */
#define EF_ARM_SYMSARESORTED        0x04
#define EF_ARM_DYNSYMSUSESEGIDX 0x08
#define EF_ARM_MAPSYMSFIRST  0x10
#define EF_ARM_EABIMASK         0XFF000000

/* Constants defined in AAELF.    */
#define EF_ARM_BE8          0x00800000
#define EF_ARM_LE8          0x00400000

#define EF_ARM_EABI_VERSION(flags)  ((flags) & EF_ARM_EABIMASK)
#define EF_ARM_EABI_UNKNOWN         0x00000000
#define EF_ARM_EABI_VER1 0x01000000
#define EF_ARM_EABI_VER2 0x02000000
#define EF_ARM_EABI_VER3 0x03000000
#define EF_ARM_EABI_VER4 0x04000000
#define EF_ARM_EABI_VER5 0x05000000
```

```
/* Additional symbol types for Thumb.  */
#define STT_ARM_TFUNC           STT_LOPROC /* A Thumb function.  */
#define STT_ARM_16BIT           STT_HIPROC /* A Thumb label.  */

/* ARM-specific values for sh_flags */
#define SHF_ARM_ENTRYSECT   0x10000000 /* Section contains an entry point */
#define SHF_ARM_COMDEF      0x80000000 /* Section may be multiply defined
                                          in the input to a link step.  */

/* ARM-specific program header flags */
#define PF_ARM_SB           0x10000000 /* Segment contains the location
                                          addressed by the static base. */
#define PF_ARM_PI           0x20000000 /* Position-independent segment.  */
#define PF_ARM_ABS          0x40000000 /* Absolute segment.  */

/* Processor specific values for the Phdr p_type field.  */
#define PT_ARM_EXIDX        (PT_LOPROC + 1)     /* ARM unwind segment.  */

/* Processor specific values for the Shdr sh_type field.  */
#define SHT_ARM_EXIDX           (SHT_LOPROC + 1) /* ARM unwind section.  */
#define SHT_ARM_PREEMPTMAP  (SHT_LOPROC + 2) /* Preemption details.  */
#define SHT_ARM_ATTRIBUTES      (SHT_LOPROC + 3) /* ARM attributes section.  */


/* AArch64 relocs.  */

#define R_AARCH64_NONE                  0 /* No relocation.  */

/* ILP32 AArch64 relocs.  */
#define R_AARCH64_P32_ABS32             1  /* Direct 32 bit.  */
#define R_AARCH64_P32_COPY          180  /* Copy symbol at runtime.  */
#define R_AARCH64_P32_GLOB_DAT          181  /* Create GOT entry.  */
#define R_AARCH64_P32_JUMP_SLOT         182  /* Create PLT entry.  */
#define R_AARCH64_P32_RELATIVE          183  /* Adjust by program base.  */
#define R_AARCH64_P32_TLS_DTPMOD        184  /* Module number, 32 bit.  */
#define R_AARCH64_P32_TLS_DTPREL 185  /* Module-relative offset, 32 bit.  */
#define R_AARCH64_P32_TLS_TPREL         186  /* TP-relative offset, 32 bit.  */
#define R_AARCH64_P32_TLSDESC           187  /* TLS Descriptor.  */
#define R_AARCH64_P32_IRELATIVE         188  /* STT_GNU_IFUNC relocation. */

/* LP64 AArch64 relocs.  */
#define R_AARCH64_ABS64             257   /* Direct 64 bit. */
#define R_AARCH64_ABS32             258   /* Direct 32 bit.  */
#define R_AARCH64_ABS16         259  /* Direct 16-bit.  */
```

```
#define R_AARCH64_PREL64 260  /* PC-relative 64-bit.      */
#define R_AARCH64_PREL32 261  /* PC-relative 32-bit.      */
#define R_AARCH64_PREL16 262  /* PC-relative 16-bit.      */
#define R_AARCH64_MOVW_UABS_G0263  /* Dir. MOVZ imm. from bits 15:0.    */
#define R_AARCH64_MOVW_UABS_G0_NC 264  /* Likewise for MOVK; no check.    */
#define R_AARCH64_MOVW_UABS_G1265  /* Dir. MOVZ imm. from bits 31:16.    */
#define R_AARCH64_MOVW_UABS_G1_NC 266  /* Likewise for MOVK; no check.    */
#define R_AARCH64_MOVW_UABS_G2267  /* Dir. MOVZ imm. from bits 47:32.    */
#define R_AARCH64_MOVW_UABS_G2_NC 268  /* Likewise for MOVK; no check.    */
#define R_AARCH64_MOVW_UABS_G3269  /* Dir. MOV{K,Z} imm. from 63:48.    */
#define R_AARCH64_MOVW_SABS_G0 270  /* Dir. MOV{N,Z} imm. from 15:0.    */
#define R_AARCH64_MOVW_SABS_G1 271  /* Dir. MOV{N,Z} imm. from 31:16.    */
#define R_AARCH64_MOVW_SABS_G2 272  /* Dir. MOV{N,Z} imm. from 47:32.    */
#define R_AARCH64_LD_PREL_LO19     273  /* PC-rel. LD imm. from bits 20:2.    */
#define R_AARCH64_ADR_PREL_LO21  274  /* PC-rel. ADR imm. from bits 20:0.    */
#define R_AARCH64_ADR_PREL_PG_HI21 275     /* Page-rel. ADRP imm. from 32:12.    */
#define R_AARCH64_ADR_PREL_PG_HI21_NC 276 /* Likewise; no overflow check.    */
#define R_AARCH64_ADD_ABS_LO12_NC 277      /* Dir. ADD imm. from bits 11:0.    */
#define R_AARCH64_LDST8_ABS_LO12_NC 278   /* Likewise for LD/ST; no check. */
#define R_AARCH64_TSTBR14       279  /* PC-rel. TBZ/TBNZ imm. from 15:2.    */
#define R_AARCH64_CONDBR19   280  /* PC-rel. cond. br. imm. from 20:2. */
#define R_AARCH64_JUMP26        282  /* PC-rel. B imm. from bits 27:2.    */
#define R_AARCH64_CALL26 283  /* Likewise for CALL.    */
#define R_AARCH64_LDST16_ABS_LO12_NC 284 /* Dir. ADD imm. from bits 11:1.    */
#define R_AARCH64_LDST32_ABS_LO12_NC 285 /* Likewise for bits 11:2.    */
#define R_AARCH64_LDST64_ABS_LO12_NC 286 /* Likewise for bits 11:3.    */
#define R_AARCH64_MOVW_PREL_G0 287  /* PC-rel. MOV{N,Z} imm. from 15:0.    */
#define R_AARCH64_MOVW_PREL_G0_NC 288   /* Likewise for MOVK; no check.    */
#define R_AARCH64_MOVW_PREL_G1 289  /* PC-rel. MOV{N,Z} imm. from 31:16. */
#define R_AARCH64_MOVW_PREL_G1_NC 290   /* Likewise for MOVK; no check.    */
#define R_AARCH64_MOVW_PREL_G2 291  /* PC-rel. MOV{N,Z} imm. from 47:32. */
#define R_AARCH64_MOVW_PREL_G2_NC 292   /* Likewise for MOVK; no check.    */
#define R_AARCH64_MOVW_PREL_G3 293  /* PC-rel. MOV{N,Z} imm. from 63:48. */
#define R_AARCH64_LDST128_ABS_LO12_NC 299 /* Dir. ADD imm. from bits 11:4.    */
#define R_AARCH64_MOVW_GOTOFF_G0 300     /* GOT-rel. off. MOV{N,Z} imm. 15:0. */
#define R_AARCH64_MOVW_GOTOFF_G0_NC 301      /* Likewise for MOVK; no check.    */
#define R_AARCH64_MOVW_GOTOFF_G1 302     /* GOT-rel. o. MOV{N,Z} imm. 31:16.    */
#define R_AARCH64_MOVW_GOTOFF_G1_NC 303      /* Likewise for MOVK; no check.    */
#define R_AARCH64_MOVW_GOTOFF_G2 304     /* GOT-rel. o. MOV{N,Z} imm. 47:32.    */
#define R_AARCH64_MOVW_GOTOFF_G2_NC 305      /* Likewise for MOVK; no check.    */
#define R_AARCH64_MOVW_GOTOFF_G3 306     /* GOT-rel. o. MOV{N,Z} imm. 63:48.    */
#define R_AARCH64_GOTREL64     307  /* GOT-relative 64-bit.    */
#define R_AARCH64_GOTREL32     308  /* GOT-relative 32-bit.    */
#define R_AARCH64_GOT_LD_PREL19  309  /* PC-rel. GOT off. load imm. 20:2.    */
```

```
#define R_AARCH64_LD64_GOTOFF_LO15 310  /* GOT-rel. off. LD/ST imm. 14:3.  */
#define R_AARCH64_ADR_GOT_PAGE  311  /* P-page-rel. GOT off. ADRP 32:12.  */
#define R_AARCH64_LD64_GOT_LO12_NC 312  /* Dir. GOT off. LD/ST imm. 11:3.  */
#define R_AARCH64_LD64_GOTPAGE_LO15 313  /* GOT-page-rel. GOT off. LD/ST 14:3 */
#define R_AARCH64_TLSGD_ADR_PREL21 512  /* PC-relative ADR imm. 20:0.  */
#define R_AARCH64_TLSGD_ADR_PAGE21 513  /* page-rel. ADRP imm. 32:12.  */
#define R_AARCH64_TLSGD_ADD_LO12_NC 514  /* direct ADD imm. from 11:0.  */
#define R_AARCH64_TLSGD_MOVW_G1  515  /* GOT-rel. MOV{N,Z} 31:16.  */
#define R_AARCH64_TLSGD_MOVW_G0_NC 516 /* GOT-rel. MOVK imm. 15:0.  */
#define R_AARCH64_TLSLD_ADR_PREL21 517  /* Like 512; local dynamic model.  */
#define R_AARCH64_TLSLD_ADR_PAGE21 518  /* Like 513; local dynamic model.  */
#define R_AARCH64_TLSLD_ADD_LO12_NC 519  /* Like 514; local dynamic model.  */
#define R_AARCH64_TLSLD_MOVW_G1  520  /* Like 515; local dynamic model.  */
#define R_AARCH64_TLSLD_MOVW_G0_NC 521  /* Like 516; local dynamic model.  */
#define R_AARCH64_TLSLD_LD_PREL19 522 /* TLS PC-rel. load imm. 20:2.  */
#define R_AARCH64_TLSLD_MOVW_DTPREL_G2 523 /* TLS DTP-rel. MOV{N,Z} 47:32.  */
#define R_AARCH64_TLSLD_MOVW_DTPREL_G1 524 /* TLS DTP-rel. MOV{N,Z} 31:16.  */
#define R_AARCH64_TLSLD_MOVW_DTPREL_G1_NC 525 /* Likewise; MOVK; no check.  */
#define R_AARCH64_TLSLD_MOVW_DTPREL_G0 526 /* TLS DTP-rel. MOV{N,Z} 15:0.  */
#define R_AARCH64_TLSLD_MOVW_DTPREL_G0_NC 527 /* Likewise; MOVK; no check.  */
#define R_AARCH64_TLSLD_ADD_DTPREL_HI12 528 /* DTP-rel. ADD imm. from 23:12. */
#define R_AARCH64_TLSLD_ADD_DTPREL_LO12 529 /* DTP-rel. ADD imm. from 11:0.  */
#define R_AARCH64_TLSLD_ADD_DTPREL_LO12_NC 530 /* Likewise; no ovfl. check.  */
#define R_AARCH64_TLSLD_LDST8_DTPREL_LO12 531 /* DTP-rel. LD/ST imm. 11:0.  */
#define R_AARCH64_TLSLD_LDST8_DTPREL_LO12_NC 532 /* Likewise; no check.  */
#define R_AARCH64_TLSLD_LDST16_DTPREL_LO12 533 /* DTP-rel. LD/ST imm. 11:1.  */
#define R_AARCH64_TLSLD_LDST16_DTPREL_LO12_NC 534 /* Likewise; no check.  */
#define R_AARCH64_TLSLD_LDST32_DTPREL_LO12 535 /* DTP-rel. LD/ST imm. 11:2.  */
#define R_AARCH64_TLSLD_LDST32_DTPREL_LO12_NC 536 /* Likewise; no check.  */
#define R_AARCH64_TLSLD_LDST64_DTPREL_LO12 537 /* DTP-rel. LD/ST imm. 11:3.  */
#define R_AARCH64_TLSLD_LDST64_DTPREL_LO12_NC 538 /* Likewise; no check.  */
#define R_AARCH64_TLSIE_MOVW_GOTTPREL_G1 539 /* GOT-rel. MOV{N,Z} 31:16.  */
#define R_AARCH64_TLSIE_MOVW_GOTTPREL_G0_NC 540 /* GOT-rel. MOVK 15:0.  */
#define R_AARCH64_TLSIE_ADR_GOTTPREL_PAGE21 541 /* Page-rel. ADRP 32:12.  */
#define R_AARCH64_TLSIE_LD64_GOTTPREL_LO12_NC 542 /* Direct LD off. 11:3.  */
#define R_AARCH64_TLSIE_LD_GOTTPREL_PREL19 543 /* PC-rel. load imm. 20:2.  */
#define R_AARCH64_TLSLE_MOVW_TPREL_G2 544 /* TLS TP-rel. MOV{N,Z} 47:32.  */
#define R_AARCH64_TLSLE_MOVW_TPREL_G1 545 /* TLS TP-rel. MOV{N,Z} 31:16.  */
#define R_AARCH64_TLSLE_MOVW_TPREL_G1_NC 546 /* Likewise; MOVK; no check.  */
#define R_AARCH64_TLSLE_MOVW_TPREL_G0 547 /* TLS TP-rel. MOV{N,Z} 15:0.  */
#define R_AARCH64_TLSLE_MOVW_TPREL_G0_NC 548 /* Likewise; MOVK; no check.  */
#define R_AARCH64_TLSLE_ADD_TPREL_HI12 549 /* TP-rel. ADD imm. 23:12.  */
#define R_AARCH64_TLSLE_ADD_TPREL_LO12 550 /* TP-rel. ADD imm. 11:0.  */
#define R_AARCH64_TLSLE_ADD_TPREL_LO12_NC 551 /* Likewise; no ovfl. check.  */
```

```
#define R_AARCH64_TLSLE_LDST8_TPREL_LO12 552 /* TP-rel. LD/ST off. 11:0.    */
#define R_AARCH64_TLSLE_LDST8_TPREL_LO12_NC 553 /* Likewise; no ovfl. check. */
#define R_AARCH64_TLSLE_LDST16_TPREL_LO12 554 /* TP-rel. LD/ST off. 11:1.    */
#define R_AARCH64_TLSLE_LDST16_TPREL_LO12_NC 555 /* Likewise; no check.    */
#define R_AARCH64_TLSLE_LDST32_TPREL_LO12 556 /* TP-rel. LD/ST off. 11:2.    */
#define R_AARCH64_TLSLE_LDST32_TPREL_LO12_NC 557 /* Likewise; no check.    */
#define R_AARCH64_TLSLE_LDST64_TPREL_LO12 558 /* TP-rel. LD/ST off. 11:3.    */
#define R_AARCH64_TLSLE_LDST64_TPREL_LO12_NC 559 /* Likewise; no check.    */
#define R_AARCH64_TLSDESC_LD_PREL19 560     /* PC-rel. load immediate 20:2.    */
#define R_AARCH64_TLSDESC_ADR_PREL21 561 /* PC-rel. ADR immediate 20:0.    */
#define R_AARCH64_TLSDESC_ADR_PAGE21 562 /* Page-rel. ADRP imm. 32:12.    */
#define R_AARCH64_TLSDESC_LD64_LO12 563     /* Direct LD off. from 11:3.    */
#define R_AARCH64_TLSDESC_ADD_LO12 564     /* Direct ADD imm. from 11:0.    */
#define R_AARCH64_TLSDESC_OFF_G1 565  /* GOT-rel. MOV{N,Z} imm. 31:16.    */
#define R_AARCH64_TLSDESC_OFF_G0_NC 566   /* GOT-rel. MOVK imm. 15:0; no ck.    */
#define R_AARCH64_TLSDESC_LDR       567  /* Relax LDR.    */
#define R_AARCH64_TLSDESC_ADD       568  /* Relax ADD.    */
#define R_AARCH64_TLSDESC_CALL      569  /* Relax BLR.    */
#define R_AARCH64_TLSLE_LDST128_TPREL_LO12 570 /* TP-rel. LD/ST off. 11:4.    */
#define R_AARCH64_TLSLE_LDST128_TPREL_LO12_NC 571 /* Likewise; no check.    */
#define R_AARCH64_TLSLD_LDST128_DTPREL_LO12 572 /* DTP-rel. LD/ST imm. 11:4. */
#define R_AARCH64_TLSLD_LDST128_DTPREL_LO12_NC 573 /* Likewise; no check.    */
#define R_AARCH64_COPY                1024  /* Copy symbol at runtime.    */
#define R_AARCH64_GLOB_DAT          1025  /* Create GOT entry.    */
#define R_AARCH64_JUMP_SLOT         1026  /* Create PLT entry.    */
#define R_AARCH64_RELATIVE          1027    /* Adjust by program base.    */
#define R_AARCH64_TLS_DTPMOD      1028 /* Module number, 64 bit.    */
#define R_AARCH64_TLS_DTPREL      1029    /* Module-relative offset, 64 bit.    */
#define R_AARCH64_TLS_TPREL       1030    /* TP-relative offset, 64 bit.    */
#define R_AARCH64_TLSDESC           1031   /* TLS Descriptor.    */
#define R_AARCH64_IRELATIVE     1032        /* STT_GNU_IFUNC relocation.    */

/* ARM relocs.    */

#define R_ARM_NONE          0     /* No reloc */
#define R_ARM_PC24          1     /* Deprecated PC relative 26
                                     bit branch.    */
#define R_ARM_ABS32         2     /* Direct 32 bit    */
#define R_ARM_REL32         3     /* PC relative 32 bit */
#define R_ARM_PC13          4
#define R_ARM_ABS16         5     /* Direct 16 bit */
#define R_ARM_ABS12         6     /* Direct 12 bit */
#define R_ARM_THM_ABS5          7     /* Direct & 0x7C (LDR, STR).    */
#define R_ARM_ABS8          8     /* Direct 8 bit */
```

```
#define R_ARM_SBREL32            9
#define R_ARM_THM_PC22           10   /* PC relative 24 bit (Thumb32 BL).    */
#define R_ARM_THM_PC8            11   /* PC relative & 0x3FC
                                 (Thumb16 LDR, ADD, ADR).    */
#define R_ARM_AMP_VCALL9         12
#define R_ARM_SWI24          13   /* Obsolete static relocation.    */
#define R_ARM_TLS_DESC           13          /* Dynamic relocation.    */
#define R_ARM_THM_SWI8           14   /* Reserved.    */
#define R_ARM_XPC25          15   /* Reserved.    */
#define R_ARM_THM_XPC22          16   /* Reserved.    */
#define R_ARM_TLS_DTPMOD32   17   /* ID of module containing symbol */
#define R_ARM_TLS_DTPOFF32   18   /* Offset in TLS block */
#define R_ARM_TLS_TPOFF32        19   /* Offset in static TLS block */
#define R_ARM_COPY           20   /* Copy symbol at runtime */
#define R_ARM_GLOB_DAT           21   /* Create GOT entry */
#define R_ARM_JUMP_SLOT          22   /* Create PLT entry */
#define R_ARM_RELATIVE           23   /* Adjust by program base */
#define R_ARM_GOTOFF             24   /* 32 bit offset to GOT */
#define R_ARM_GOTPC          25   /* 32 bit PC relative offset to GOT */
#define R_ARM_GOT32          26   /* 32 bit GOT entry */
#define R_ARM_PLT32          27   /* Deprecated, 32 bit PLT address.    */
#define R_ARM_CALL           28   /* PC relative 24 bit (BL, BLX).    */
#define R_ARM_JUMP24             29   /* PC relative 24 bit
                                 (B, BL<cond>).    */
#define R_ARM_THM_JUMP24     30   /* PC relative 24 bit (Thumb32 B.W).    */
#define R_ARM_BASE_ABS           31   /* Adjust by program base.    */
#define R_ARM_ALU_PCREL_7_0  32   /* Obsolete.    */
#define R_ARM_ALU_PCREL_15_8 33   /* Obsolete.    */
#define R_ARM_ALU_PCREL_23_15    34   /* Obsolete.    */
#define R_ARM_LDR_SBREL_11_0 35   /* Deprecated, prog. base relative.    */
#define R_ARM_ALU_SBREL_19_12    36   /* Deprecated, prog. base relative.    */
#define R_ARM_ALU_SBREL_27_20    37   /* Deprecated, prog. base relative.    */
#define R_ARM_TARGET1            38
#define R_ARM_SBREL31            39   /* Program base relative.    */
#define R_ARM_V4BX           40
#define R_ARM_TARGET2            41
#define R_ARM_PREL31         42   /* 32 bit PC relative.    */
#define R_ARM_MOVW_ABS_NC    43   /* Direct 16-bit (MOVW).    */
#define R_ARM_MOVT_ABS           44   /* Direct high 16-bit (MOVT).    */
#define R_ARM_MOVW_PREL_NC   45   /* PC relative 16-bit (MOVW).    */
#define R_ARM_MOVT_PREL          46   /* PC relative (MOVT).    */
#define R_ARM_THM_MOVW_ABS_NC47   /* Direct 16 bit (Thumb32 MOVW).    */
#define R_ARM_THM_MOVT_ABS   48   /* Direct high 16 bit
                                 (Thumb32 MOVT).    */
```

```
#define R_ARM_THM_MOVW_PREL_NC      49    /* PC relative 16 bit
                                    (Thumb32 MOVW).    */
#define R_ARM_THM_MOVT_PREL         50    /* PC relative high 16 bit
                                    (Thumb32 MOVT).    */
#define R_ARM_THM_JUMP19            51    /* PC relative 20 bit
                                    (Thumb32 B<cond>.W).    */
#define R_ARM_THM_JUMP6             52    /* PC relative X & 0x7E
                                    (Thumb16 CBZ, CBNZ).    */
#define R_ARM_THM_ALU_PREL_11_0 53    /* PC relative 12 bit
                                    (Thumb32 ADR.W).    */
#define R_ARM_THM_PC12              54    /* PC relative 12 bit
                                    (Thumb32 LDR{D,SB,H,SH}).    */
#define R_ARM_ABS32_NOI             55    /* Direct 32-bit.    */
#define R_ARM_REL32_NOI             56    /* PC relative 32-bit.    */
#define R_ARM_ALU_PC_G0_NC          57    /* PC relative (ADD, SUB).    */
#define R_ARM_ALU_PC_G0             58    /* PC relative (ADD, SUB).    */
#define R_ARM_ALU_PC_G1_NC          59    /* PC relative (ADD, SUB).    */
#define R_ARM_ALU_PC_G1             60    /* PC relative (ADD, SUB).    */
#define R_ARM_ALU_PC_G2             61    /* PC relative (ADD, SUB).    */
#define R_ARM_LDR_PC_G1             62    /* PC relative (LDR,STR,LDRB,STRB).    */
#define R_ARM_LDR_PC_G2             63    /* PC relative (LDR,STR,LDRB,STRB).    */
#define R_ARM_LDRS_PC_G0            64    /* PC relative (STR{D,H},
                                    LDR{D,SB,H,SH}).    */
#define R_ARM_LDRS_PC_G1            65    /* PC relative (STR{D,H},
                                    LDR{D,SB,H,SH}).    */
#define R_ARM_LDRS_PC_G2            66    /* PC relative (STR{D,H},
                                    LDR{D,SB,H,SH}).    */
#define R_ARM_LDC_PC_G0             67    /* PC relative (LDC, STC).    */
#define R_ARM_LDC_PC_G1             68    /* PC relative (LDC, STC).    */
#define R_ARM_LDC_PC_G2             69    /* PC relative (LDC, STC).    */
#define R_ARM_ALU_SB_G0_NC          70    /* Program base relative (ADD,SUB).    */
#define R_ARM_ALU_SB_G0             71    /* Program base relative (ADD,SUB).    */
#define R_ARM_ALU_SB_G1_NC          72    /* Program base relative (ADD,SUB).    */
#define R_ARM_ALU_SB_G1             73    /* Program base relative (ADD,SUB).    */
#define R_ARM_ALU_SB_G2             74    /* Program base relative (ADD,SUB).    */
#define R_ARM_LDR_SB_G0             75    /* Program base relative (LDR,
                                    STR, LDRB, STRB).    */
#define R_ARM_LDR_SB_G1             76    /* Program base relative
                                    (LDR, STR, LDRB, STRB).    */
#define R_ARM_LDR_SB_G2             77    /* Program base relative
                                    (LDR, STR, LDRB, STRB).    */
#define R_ARM_LDRS_SB_G078    /* Program base relative
                                    (LDR, STR, LDRB, STRB).    */
#define R_ARM_LDRS_SB_G179    /* Program base relative
```

```
                                (LDR, STR, LDRB, STRB).    */
#define R_ARM_LDRS_SB_G280    /* Program base relative
                                (LDR, STR, LDRB, STRB).    */
#define R_ARM_LDC_SB_G0         81    /* Program base relative (LDC,STC).    */
#define R_ARM_LDC_SB_G1         82    /* Program base relative (LDC,STC).    */
#define R_ARM_LDC_SB_G2         83    /* Program base relative (LDC,STC).    */
#define R_ARM_MOVW_BREL_NC84    /* Program base relative 16
                                bit (MOVW).    */
#define R_ARM_MOVT_BREL         85    /* Program base relative high
                                16 bit (MOVT).    */
#define R_ARM_MOVW_BREL             86    /* Program base relative 16
                                bit (MOVW).    */
#define R_ARM_THM_MOVW_BREL_NC      87    /* Program base relative 16
                                bit (Thumb32 MOVW).    */
#define R_ARM_THM_MOVT_BREL      88    /* Program base relative high
                                16 bit (Thumb32 MOVT).    */
#define R_ARM_THM_MOVW_BREL     89    /* Program base relative 16
                                bit (Thumb32 MOVW).    */
#define R_ARM_TLS_GOTDESC      90
#define R_ARM_TLS_CALL          91
#define R_ARM_TLS_DESCSEQ      92    /* TLS relaxation.    */
#define R_ARM_THM_TLS_CALL    93
#define R_ARM_PLT32_ABS         94
#define R_ARM_GOT_ABS           95    /* GOT entry.    */
#define R_ARM_GOT_PREL          96    /* PC relative GOT entry.    */
#define R_ARM_GOT_BREL12        97    /* GOT entry relative to GOT
                                origin (LDR).    */
#define R_ARM_GOTOFF12          98    /* 12 bit, GOT entry relative
                                to GOT origin (LDR, STR).    */
#define R_ARM_GOTRELAX          99
#define R_ARM_GNU_VTENTRY    100
#define R_ARM_GNU_VTINHERIT  101
#define R_ARM_THM_PC11          102  /* PC relative & 0xFFE (Thumb16 B).    */
#define R_ARM_THM_PC9           103  /* PC relative & 0x1FE
                                (Thumb16 B/B<cond>).    */
#define R_ARM_TLS_GD32          104  /* PC-rel 32 bit for global dynamic
                                thread local data */
#define R_ARM_TLS_LDM32         105  /* PC-rel 32 bit for local dynamic
                                thread local data */
#define R_ARM_TLS_LDO32         106  /* 32 bit offset relative to TLS
                                block */
#define R_ARM_TLS_IE32          107  /* PC-rel 32 bit for GOT entry of
                                static TLS block offset */
#define R_ARM_TLS_LE32          108  /* 32 bit offset relative to static
```

```
                                TLS block */
#define R_ARM_TLS_LDO12         109  /* 12 bit relative to TLS
                                block (LDR, STR).    */
#define R_ARM_TLS_LE12          110  /* 12 bit relative to static
                                TLS block (LDR, STR).    */
#define R_ARM_TLS_IE12GP  111  /* 12 bit GOT entry relative
                                to GOT origin (LDR).    */
#define R_ARM_ME_TOO            128  /* Obsolete.    */
#define R_ARM_THM_TLS_DESCSEQ    129
#define R_ARM_THM_TLS_DESCSEQ16 129
#define R_ARM_THM_TLS_DESCSEQ32 130
#define R_ARM_THM_GOT_BREL12     131  /* GOT entry relative to GOT
                                origin, 12 bit (Thumb32 LDR).    */
#define R_ARM_IRELATIVE          160
#define R_ARM_RXPC25       249
#define R_ARM_RSBREL32          250
#define R_ARM_THM_RPC22       251
#define R_ARM_RREL32       252
#define R_ARM_RABS22       253
#define R_ARM_RPC24        254
#define R_ARM_RBASE        255
/* Keep this the last entry.    */
#define R_ARM_NUM          256

/* IA-64 specific declarations.    */

/* Processor specific flags for the Ehdr e_flags field.    */
#define EF_IA_64_MASKOS         0x0000000f    /* os-specific flags */
#define EF_IA_64_ABI64       0x00000010    /* 64-bit ABI */
#define EF_IA_64_ARCH       0xff000000     /* arch. version mask */

/* Processor specific values for the Phdr p_type field.    */
#define PT_IA_64_ARCHEXT  (PT_LOPROC + 0)    /* arch extension bits */
#define PT_IA_64_UNWIND         (PT_LOPROC + 1)    /* ia64 unwind bits */
#define PT_IA_64_HP_OPT_ANOT (PT_LOOS + 0x12)
#define PT_IA_64_HP_HSL_ANOT  (PT_LOOS + 0x13)
#define PT_IA_64_HP_STACK (PT_LOOS + 0x14)

/* Processor specific flags for the Phdr p_flags field.    */
#define PF_IA_64_NORECOV 0x80000000    /* spec insns w/o recovery */

/* Processor specific values for the Shdr sh_type field.    */
#define SHT_IA_64_EXT       (SHT_LOPROC + 0) /* extension bits */
#define SHT_IA_64_UNWIND (SHT_LOPROC + 1) /* unwind bits */
```

```
/* Processor specific flags for the Shdr sh_flags field.  */
#define SHF_IA_64_SHORT         0x10000000  /* section near gp */
#define SHF_IA_64_NORECOV       0x20000000  /* spec insns w/o recovery */

/* Processor specific values for the Dyn d_tag field.  */
#define DT_IA_64_PLT_RESERVE  (DT_LOPROC + 0)
#define DT_IA_64_NUM        1

/* IA-64 relocations.  */
#define R_IA64_NONE         0x00/* none */
#define R_IA64_IMM14        0x21/* symbol + addend, add imm14 */
#define R_IA64_IMM22        0x22/* symbol + addend, add imm22 */
#define R_IA64_IMM64        0x23/* symbol + addend, mov imm64 */
#define R_IA64_DIR32MSB         0x24/* symbol + addend, data4 MSB */
#define R_IA64_DIR32LSB         0x25/* symbol + addend, data4 LSB */
#define R_IA64_DIR64MSB         0x26/* symbol + addend, data8 MSB */
#define R_IA64_DIR64LSB         0x27/* symbol + addend, data8 LSB */
#define R_IA64_GPREL22          0x2a/* @gprel(sym + add), add imm22 */
#define R_IA64_GPREL64I         0x2b/* @gprel(sym + add), mov imm64 */
#define R_IA64_GPREL32MSB       0x2c/* @gprel(sym + add), data4 MSB */
#define R_IA64_GPREL32LSB 0x2d/* @gprel(sym + add), data4 LSB */
#define R_IA64_GPREL64MSB       0x2e/* @gprel(sym + add), data8 MSB */
#define R_IA64_GPREL64LSB 0x2f /* @gprel(sym + add), data8 LSB */
#define R_IA64_LTOFF22          0x32/* @ltoff(sym + add), add imm22 */
#define R_IA64_LTOFF64I         0x33/* @ltoff(sym + add), mov imm64 */
#define R_IA64_PLTOFF22         0x3a/* @pltoff(sym + add), add imm22 */
#define R_IA64_PLTOFF64I    0x3b/* @pltoff(sym + add), mov imm64 */
#define R_IA64_PLTOFF64MSB      0x3e/* @pltoff(sym + add), data8 MSB */
#define R_IA64_PLTOFF64LSB      0x3f /* @pltoff(sym + add), data8 LSB */
#define R_IA64_FPTR64I      0x43/* @fptr(sym + add), mov imm64 */
#define R_IA64_FPTR32MSB  0x44/* @fptr(sym + add), data4 MSB */
#define R_IA64_FPTR32LSB    0x45/* @fptr(sym + add), data4 LSB */
#define R_IA64_FPTR64MSB  0x46/* @fptr(sym + add), data8 MSB */
#define R_IA64_FPTR64LSB    0x47/* @fptr(sym + add), data8 LSB */
#define R_IA64_PCREL60B         0x48/* @pcrel(sym + add), brl */
#define R_IA64_PCREL21B         0x49/* @pcrel(sym + add), ptb, call */
#define R_IA64_PCREL21M         0x4a/* @pcrel(sym + add), chk.s */
#define R_IA64_PCREL21F         0x4b/* @pcrel(sym + add), fchkf */
#define R_IA64_PCREL32MSB       0x4c/* @pcrel(sym + add), data4 MSB */
#define R_IA64_PCREL32LSB  0x4d/* @pcrel(sym + add), data4 LSB */
#define R_IA64_PCREL64MSB       0x4e/* @pcrel(sym + add), data8 MSB */
#define R_IA64_PCREL64LSB  0x4f /* @pcrel(sym + add), data8 LSB */
#define R_IA64_LTOFF_FPTR22     0x52/* @ltoff(@fptr(s+a)), imm22 */
```

```
#define R_IA64_LTOFF_FPTR64I    0x53/* @ltoff(@fptr(s+a)), imm64 */
#define R_IA64_LTOFF_FPTR32MSB      0x54/* @ltoff(@fptr(s+a)), data4 MSB */
#define R_IA64_LTOFF_FPTR32LSB      0x55/* @ltoff(@fptr(s+a)), data4 LSB */
#define R_IA64_LTOFF_FPTR64MSB      0x56/* @ltoff(@fptr(s+a)), data8 MSB */
#define R_IA64_LTOFF_FPTR64LSB      0x57/* @ltoff(@fptr(s+a)), data8 LSB */
#define R_IA64_SEGREL32MSB      0x5c/* @segrel(sym + add), data4 MSB */
#define R_IA64_SEGREL32LSB      0x5d/* @segrel(sym + add), data4 LSB */
#define R_IA64_SEGREL64MSB      0x5e/* @segrel(sym + add), data8 MSB */
#define R_IA64_SEGREL64LSB      0x5f /* @segrel(sym + add), data8 LSB */
#define R_IA64_SECREL32MSB      0x64/* @secrel(sym + add), data4 MSB */
#define R_IA64_SECREL32LSB      0x65/* @secrel(sym + add), data4 LSB */
#define R_IA64_SECREL64MSB      0x66/* @secrel(sym + add), data8 MSB */
#define R_IA64_SECREL64LSB      0x67/* @secrel(sym + add), data8 LSB */
#define R_IA64_REL32MSB         0x6c/* data 4 + REL */
#define R_IA64_REL32LSB         0x6d/* data 4 + REL */
#define R_IA64_REL64MSB         0x6e/* data 8 + REL */
#define R_IA64_REL64LSB         0x6f /* data 8 + REL */
#define R_IA64_LTV32MSB         0x74/* symbol + addend, data4 MSB */
#define R_IA64_LTV32LSB         0x75/* symbol + addend, data4 LSB */
#define R_IA64_LTV64MSB         0x76/* symbol + addend, data8 MSB */
#define R_IA64_LTV64LSB         0x77/* symbol + addend, data8 LSB */
#define R_IA64_PCREL21BI    0x79/* @pcrel(sym + add), 21bit inst */
#define R_IA64_PCREL22          0x7a/* @pcrel(sym + add), 22bit inst */
#define R_IA64_PCREL64I         0x7b/* @pcrel(sym + add), 64bit inst */
#define R_IA64_IPLTMSB          0x80/* dynamic reloc, imported PLT, MSB */
#define R_IA64_IPLTLSB      0x81/* dynamic reloc, imported PLT, LSB */
#define R_IA64_COPY         0x84/* copy relocation */
#define R_IA64_SUB          0x85/* Addend and symbol difference */
#define R_IA64_LTOFF22X         0x86/* LTOFF22, relaxable.    */
#define R_IA64_LDXMOV           0x87/* Use of LTOFF22X.    */
#define R_IA64_TPREL14          0x91/* @tprel(sym + add), imm14 */
#define R_IA64_TPREL22          0x92/* @tprel(sym + add), imm22 */
#define R_IA64_TPREL64I         0x93/* @tprel(sym + add), imm64 */
#define R_IA64_TPREL64MSB       0x96/* @tprel(sym + add), data8 MSB */
#define R_IA64_TPREL64LSB  0x97/* @tprel(sym + add), data8 LSB */
#define R_IA64_LTOFF_TPREL22    0x9a/* @ltoff(@tprel(s+a)), imm2 */
#define R_IA64_DTPMOD64MSB   0xa6/* @dtpmod(sym + add), data8 MSB */
#define R_IA64_DTPMOD64LSB    0xa7/* @dtpmod(sym + add), data8 LSB */
#define R_IA64_LTOFF_DTPMOD22       0xaa/* @ltoff(@dtpmod(sym + add)), imm22 */
#define R_IA64_DTPREL14         0xb1/* @dtprel(sym + add), imm14 */
#define R_IA64_DTPREL22         0xb2/* @dtprel(sym + add), imm22 */
#define R_IA64_DTPREL64I    0xb3/* @dtprel(sym + add), imm64 */
#define R_IA64_DTPREL32MSB      0xb4/* @dtprel(sym + add), data4 MSB */
#define R_IA64_DTPREL32LSB      0xb5/* @dtprel(sym + add), data4 LSB */
```

```
#define R_IA64_DTPREL64MSB     0xb6/* @dtprel(sym + add), data8 MSB */
#define R_IA64_DTPREL64LSB     0xb7/* @dtprel(sym + add), data8 LSB */
#define R_IA64_LTOFF_DTPREL22  0xba/* @ltoff(@dtprel(s+a)), imm22 */

/* SH specific declarations */

/* Processor specific flags for the ELF header e_flags field.    */
#define EF_SH_MACH_MASK        0x1f
#define EF_SH_UNKNOWN          0x0
#define EF_SH1                 0x1
#define EF_SH2                 0x2
#define EF_SH3                 0x3
#define EF_SH_DSP       0x4
#define EF_SH3_DSP             0x5
#define EF_SH4AL_DSP           0x6
#define EF_SH3E                0x8
#define EF_SH4                 0x9
#define EF_SH2E                0xb
#define EF_SH4A                0xc
#define EF_SH2A                0xd
#define EF_SH4_NOFPU           0x10
#define EF_SH4A_NOFPU          0x11
#define EF_SH4_NOMMU_NOFPU     0x12
#define EF_SH2A_NOFPU          0x13
#define EF_SH3_NOMMU           0x14
#define EF_SH2A_SH4_NOFPU      0x15
#define EF_SH2A_SH3_NOFPU      0x16
#define EF_SH2A_SH4            0x17
#define EF_SH2A_SH3E           0x18

/* SH relocs.    */
#define    R_SH_NONE           0
#define    R_SH_DIR32          1
#define    R_SH_REL32          2
#define    R_SH_DIR8WPN        3
#define    R_SH_IND12W         4
#define    R_SH_DIR8WPL        5
#define    R_SH_DIR8WPZ        6
#define    R_SH_DIR8BP         7
#define    R_SH_DIR8W          8
#define    R_SH_DIR8L          9
#define    R_SH_SWITCH16       25
#define    R_SH_SWITCH32       26
#define    R_SH_USES           27
```

```
#define	R_SH_COUNT		28
#define	R_SH_ALIGN		29
#define	R_SH_CODE		30
#define	R_SH_DATA		31
#define	R_SH_LABEL		32
#define	R_SH_SWITCH8		33
#define	R_SH_GNU_VTINHERIT	34
#define	R_SH_GNU_VTENTRY	35
#define	R_SH_TLS_GD_32		144
#define	R_SH_TLS_LD_32		145
#define	R_SH_TLS_LDO_32		146
#define	R_SH_TLS_IE_32		147
#define	R_SH_TLS_LE_32		148
#define	R_SH_TLS_DTPMOD32	149
#define	R_SH_TLS_DTPOFF32	150
#define	R_SH_TLS_TPOFF32	151
#define	R_SH_GOT32		160
#define	R_SH_PLT32		161
#define	R_SH_COPY		162
#define	R_SH_GLOB_DAT		163
#define	R_SH_JMP_SLOT		164
#define	R_SH_RELATIVE		165
#define	R_SH_GOTOFF		166
#define	R_SH_GOTPC		167
/* Keep this the last entry.  */
#define	R_SH_NUM		256

/* S/390 specific definitions.  */

/* Valid values for the e_flags field.  */

#define EF_S390_HIGH_GPRS	0x00000001	/* High GPRs kernel facility needed.  */

/* Additional s390 relocs */

#define R_390_NONE		0	/* No reloc.  */
#define R_390_8			1	/* Direct 8 bit.  */
#define R_390_12		2	/* Direct 12 bit.  */
#define R_390_16		3	/* Direct 16 bit.  */
#define R_390_32		4	/* Direct 32 bit.  */
#define R_390_PC32		5	/* PC relative 32 bit.	*/
#define R_390_GOT12		6	/* 12 bit GOT offset.  */
#define R_390_GOT32		7	/* 32 bit GOT offset.  */
#define R_390_PLT32		8	/* 32 bit PC relative PLT address.  */
```

```
#define R_390_COPY          9      /* Copy symbol at runtime.    */
#define R_390_GLOB_DAT          10    /* Create GOT entry.    */
#define R_390_JMP_SLOT          11    /* Create PLT entry.    */
#define R_390_RELATIVE          12    /* Adjust by program base.    */
#define R_390_GOTOFF32          13    /* 32 bit offset to GOT.     */
#define R_390_GOTPC         14    /* 32 bit PC relative offset to GOT.    */
#define R_390_GOT16         15    /* 16 bit GOT offset.    */
#define R_390_PC16          16    /* PC relative 16 bit.       */
#define R_390_PC16DBL       17    /* PC relative 16 bit shifted by 1.    */
#define R_390_PLT16DBL          18    /* 16 bit PC rel. PLT shifted by 1.    */
#define R_390_PC32DBL       19    /* PC relative 32 bit shifted by 1.    */
#define R_390_PLT32DBL          20    /* 32 bit PC rel. PLT shifted by 1.    */
#define R_390_GOTPCDBL          21    /* 32 bit PC rel. GOT shifted by 1.    */
#define R_390_64        22  /* Direct 64 bit.    */
#define R_390_PC64          23    /* PC relative 64 bit.       */
#define R_390_GOT64         24    /* 64 bit GOT offset.    */
#define R_390_PLT64         25    /* 64 bit PC relative PLT address.    */
#define R_390_GOTENT        26    /* 32 bit PC rel. to GOT entry >> 1. */
#define R_390_GOTOFF16          27    /* 16 bit offset to GOT. */
#define R_390_GOTOFF64          28    /* 64 bit offset to GOT. */
#define R_390_GOTPLT12          29    /* 12 bit offset to jump slot.   */
#define R_390_GOTPLT16          30    /* 16 bit offset to jump slot.   */
#define R_390_GOTPLT32          31    /* 32 bit offset to jump slot.   */
#define R_390_GOTPLT64          32    /* 64 bit offset to jump slot.   */
#define R_390_GOTPLTENT         33    /* 32 bit rel. offset to jump slot.    */
#define R_390_PLTOFF16          34    /* 16 bit offset from GOT to PLT. */
#define R_390_PLTOFF32          35    /* 32 bit offset from GOT to PLT. */
#define R_390_PLTOFF64          36    /* 16 bit offset from GOT to PLT. */
#define R_390_TLS_LOAD          37    /* Tag for load insn in TLS code.    */
#define R_390_TLS_GDCALL  38    /* Tag for function call in general
                                 dynamic TLS code. */
#define R_390_TLS_LDCALL   39    /* Tag for function call in local
                                 dynamic TLS code. */
#define R_390_TLS_GD32          40    /* Direct 32 bit for general dynamic
                                 thread local data.    */
#define R_390_TLS_GD64          41    /* Direct 64 bit for general dynamic
                               thread local data.    */
#define R_390_TLS_GOTIE12 42    /* 12 bit GOT offset for static TLS
                                 block offset.    */
#define R_390_TLS_GOTIE32 43    /* 32 bit GOT offset for static TLS
                                 block offset.    */
#define R_390_TLS_GOTIE64 44    /* 64 bit GOT offset for static TLS
                                 block offset. */
#define R_390_TLS_LDM32         45    /* Direct 32 bit for local dynamic
```

```
                        thread local data in LE code.    */
#define R_390_TLS_LDM64         46    /* Direct 64 bit for local dynamic
                        thread local data in LE code.    */
#define R_390_TLS_IE32      47    /* 32 bit address of GOT entry for
                        negated static TLS block offset.    */
#define R_390_TLS_IE64      48    /* 64 bit address of GOT entry for
                        negated static TLS block offset.    */
#define R_390_TLS_IEENT         49    /* 32 bit rel. offset to GOT entry for
                        negated static TLS block offset.    */
#define R_390_TLS_LE32          50    /* 32 bit negated offset relative to
                        static TLS block.    */
#define R_390_TLS_LE64          51    /* 64 bit negated offset relative to
                        static TLS block.    */
#define R_390_TLS_LDO32         52    /* 32 bit offset relative to TLS
                        block.    */
#define R_390_TLS_LDO64         53    /* 64 bit offset relative to TLS
                        block.    */
#define R_390_TLS_DTPMOD54    /* ID of module containing symbol.    */
#define R_390_TLS_DTPOFF  55    /* Offset in TLS block.       */
#define R_390_TLS_TPOFF         56    /* Negated offset in static TLS
                        block.    */
#define R_390_20        57    /* Direct 20 bit.    */
#define R_390_GOT20         58    /* 20 bit GOT offset.    */
#define R_390_GOTPLT20          59    /* 20 bit offset to jump slot.    */
#define R_390_TLS_GOTIE20 60    /* 20 bit GOT offset for static TLS
                        block offset.    */
#define R_390_IRELATIVE             61          /* STT_GNU_IFUNC relocation.    */
/* Keep this the last entry.    */
#define R_390_NUM           62


/* CRIS relocations.    */
#define R_CRIS_NONE         0
#define R_CRIS_8        1
#define R_CRIS_16       2
#define R_CRIS_32       3
#define R_CRIS_8_PCREL          4
#define R_CRIS_16_PCREL         5
#define R_CRIS_32_PCREL         6
#define R_CRIS_GNU_VTINHERIT  7
#define R_CRIS_GNU_VTENTRY    8
#define R_CRIS_COPY         9
#define R_CRIS_GLOB_DAT         10
#define R_CRIS_JUMP_SLOT  11
```

```
#define R_CRIS_RELATIVE         12
#define R_CRIS_16_GOT       13
#define R_CRIS_32_GOT       14
#define R_CRIS_16_GOTPLT    15
#define R_CRIS_32_GOTPLT    16
#define R_CRIS_32_GOTREL    17
#define R_CRIS_32_PLT_GOTREL    18
#define R_CRIS_32_PLT_PCREL     19

#define R_CRIS_NUM          20


/* AMD x86-64 relocations.  */
#define R_X86_64_NONE           0   /* No reloc */
#define R_X86_64_64         1   /* Direct 64 bit  */
#define R_X86_64_PC32       2   /* PC relative 32 bit signed */
#define R_X86_64_GOT32          3   /* 32 bit GOT entry */
#define R_X86_64_PLT32          4   /* 32 bit PLT address */
#define R_X86_64_COPY           5   /* Copy symbol at runtime */
#define R_X86_64_GLOB_DAT       6   /* Create GOT entry */
#define R_X86_64_JUMP_SLOT      7   /* Create PLT entry */
#define R_X86_64_RELATIVE   8   /* Adjust by program base */
#define R_X86_64_GOTPCREL       9   /* 32 bit signed PC relative
                                offset to GOT */
#define R_X86_64_32         10  /* Direct 32 bit zero extended */
#define R_X86_64_32S        11  /* Direct 32 bit sign extended */
#define R_X86_64_16         12  /* Direct 16 bit zero extended */
#define R_X86_64_PC16       13  /* 16 bit sign extended pc relative */
#define R_X86_64_8          14  /* Direct 8 bit sign extended  */
#define R_X86_64_PC8        15  /* 8 bit sign extended pc relative */
#define R_X86_64_DTPMOD64   16  /* ID of module containing symbol */
#define R_X86_64_DTPOFF64       17  /* Offset in module's TLS block */
#define R_X86_64_TPOFF64    18  /* Offset in initial TLS block */
#define R_X86_64_TLSGD          19  /* 32 bit signed PC relative offset
                                to two GOT entries for GD symbol */
#define R_X86_64_TLSLD          20  /* 32 bit signed PC relative offset
                                to two GOT entries for LD symbol */
#define R_X86_64_DTPOFF32       21  /* Offset in TLS block */
#define R_X86_64_GOTTPOFF       22  /* 32 bit signed PC relative offset
                                to GOT entry for IE symbol */
#define R_X86_64_TPOFF32    23  /* Offset in initial TLS block */
#define R_X86_64_PC64       24  /* PC relative 64 bit */
#define R_X86_64_GOTOFF64       25  /* 64 bit offset to GOT */
#define R_X86_64_GOTPC32    26  /* 32 bit signed pc relative
```

```
                                offset to GOT */
#define R_X86_64_GOT64          27    /* 64-bit GOT entry offset */
#define R_X86_64_GOTPCREL64     28    /* 64-bit PC relative offset
                                to GOT entry */
#define R_X86_64_GOTPC64 29     /* 64-bit PC relative offset to GOT */
#define R_X86_64_GOTPLT6430     /* like GOT64, says PLT entry needed */
#define R_X86_64_PLTOFF64 31    /* 64-bit GOT relative offset
                                to PLT entry */
#define R_X86_64_SIZE32         32    /* Size of symbol plus 32-bit addend */
#define R_X86_64_SIZE64         33    /* Size of symbol plus 64-bit addend */
#define R_X86_64_GOTPC32_TLSDESC 34     /* GOT offset for TLS descriptor.    */
#define R_X86_64_TLSDESC_CALL      35       /* Marker for call through TLS
                                descriptor.    */
#define R_X86_64_TLSDESC            36      /* TLS descriptor.    */
#define R_X86_64_IRELATIVE 37     /* Adjust indirectly by program base */
#define R_X86_64_RELATIVE64     38    /* 64-bit adjust by program base */

#define R_X86_64_NUM            39


/* AM33 relocations.    */
#define R_MN10300_NONE          0     /* No reloc.    */
#define R_MN10300_32        1     /* Direct 32 bit.    */
#define R_MN10300_16        2     /* Direct 16 bit.    */
#define R_MN10300_8         3     /* Direct 8 bit.    */
#define R_MN10300_PCREL32       4     /* PC-relative 32-bit.    */
#define R_MN10300_PCREL16       5     /* PC-relative 16-bit signed.    */
#define R_MN10300_PCREL8 6      /* PC-relative 8-bit signed.    */
#define R_MN10300_GNU_VTINHERIT  7      /* Ancient C++ vtable garbage... */
#define R_MN10300_GNU_VTENTRY    8      /* ... collection annotation.    */
#define R_MN10300_24        9     /* Direct 24 bit.    */
#define R_MN10300_GOTPC32       10    /* 32-bit PCrel offset to GOT.    */
#define R_MN10300_GOTPC16       11    /* 16-bit PCrel offset to GOT.    */
#define R_MN10300_GOTOFF32      12    /* 32-bit offset from GOT.    */
#define R_MN10300_GOTOFF24      13    /* 24-bit offset from GOT.    */
#define R_MN10300_GOTOFF16      14    /* 16-bit offset from GOT.    */
#define R_MN10300_PLT32         15    /* 32-bit PCrel to PLT entry.    */
#define R_MN10300_PLT16         16    /* 16-bit PCrel to PLT entry.    */
#define R_MN10300_GOT32         17    /* 32-bit offset to GOT entry.    */
#define R_MN10300_GOT24         18    /* 24-bit offset to GOT entry.    */
#define R_MN10300_GOT16         19    /* 16-bit offset to GOT entry.    */
#define R_MN10300_COPY          20    /* Copy symbol at runtime.    */
#define R_MN10300_GLOB_DAT      21    /* Create GOT entry.    */
#define R_MN10300_JMP_SLOT      22    /* Create PLT entry.    */
```

```
#define R_MN10300_RELATIVE      23    /* Adjust by program base.    */
#define R_MN10300_TLS_GD24    /* 32-bit offset for global dynamic.    */
#define R_MN10300_TLS_LD 25    /* 32-bit offset for local dynamic.    */
#define R_MN10300_TLS_LDO      26    /* Module-relative offset.    */
#define R_MN10300_TLS_GOTIE   27    /* GOT offset for static TLS block
                                offset.    */
#define R_MN10300_TLS_IE  28    /* GOT address for static TLS block
                                offset.    */
#define R_MN10300_TLS_LE  29    /* Offset relative to static TLS
                                block.    */
#define R_MN10300_TLS_DTPMOD      30    /* ID of module containing symbol.    */
#define R_MN10300_TLS_DTPOFF 31    /* Offset in module TLS block.    */
#define R_MN10300_TLS_TPOFF   32    /* Offset in static TLS block.    */
#define R_MN10300_SYM_DIFF    33    /* Adjustment for next reloc as needed
                                by linker relaxation.    */
#define R_MN10300_ALIGN         34    /* Alignment requirement for linker
                                relaxation.    */
#define R_MN10300_NUM           35


/* M32R relocs.    */
#define R_M32R_NONE         0     /* No reloc. */
#define R_M32R_16           1     /* Direct 16 bit. */
#define R_M32R_32           2     /* Direct 32 bit. */
#define R_M32R_24           3     /* Direct 24 bit. */
#define R_M32R_10_PCREL         4     /* PC relative 10 bit shifted. */
#define R_M32R_18_PCREL         5     /* PC relative 18 bit shifted. */
#define R_M32R_26_PCREL         6     /* PC relative 26 bit shifted. */
#define R_M32R_HI16_ULO         7     /* High 16 bit with unsigned low. */
#define R_M32R_HI16_SLO         8     /* High 16 bit with signed low. */
#define R_M32R_LO16         9     /* Low 16 bit. */
#define R_M32R_SDA16        10    /* 16 bit offset in SDA. */
#define R_M32R_GNU_VTINHERIT        11
#define R_M32R_GNU_VTENTRY  12
/* M32R relocs use SHT_RELA.    */
#define R_M32R_16_RELA          33    /* Direct 16 bit. */
#define R_M32R_32_RELA          34    /* Direct 32 bit. */
#define R_M32R_24_RELA          35    /* Direct 24 bit. */
#define R_M32R_10_PCREL_RELA 36    /* PC relative 10 bit shifted. */
#define R_M32R_18_PCREL_RELA 37    /* PC relative 18 bit shifted. */
#define R_M32R_26_PCREL_RELA 38    /* PC relative 26 bit shifted. */
#define R_M32R_HI16_ULO_RELA 39    /* High 16 bit with unsigned low */
#define R_M32R_HI16_SLO_RELA  40    /* High 16 bit with signed low */
#define R_M32R_LO16_RELA 41    /* Low 16 bit */
```

```
#define R_M32R_SDA16_RELA      42    /* 16 bit offset in SDA */
#define R_M32R_RELA_GNU_VTINHERIT      43
#define R_M32R_RELA_GNU_VTENTRY 44
#define R_M32R_REL32        45    /* PC relative 32 bit.    */

#define R_M32R_GOT24            48    /* 24 bit GOT entry */
#define R_M32R_26_PLTREL  49    /* 26 bit PC relative to PLT shifted */
#define R_M32R_COPY         50    /* Copy symbol at runtime */
#define R_M32R_GLOB_DAT         51    /* Create GOT entry */
#define R_M32R_JMP_SLOT         52    /* Create PLT entry */
#define R_M32R_RELATIVE         53    /* Adjust by program base */
#define R_M32R_GOTOFF           54    /* 24 bit offset to GOT */
#define R_M32R_GOTPC24          55    /* 24 bit PC relative offset to GOT */
#define R_M32R_GOT16_HI_ULO  56    /* High 16 bit GOT entry with unsigned
                                low */
#define R_M32R_GOT16_HI_SLO  57    /* High 16 bit GOT entry with signed
                                low */
#define R_M32R_GOT16_LO         58    /* Low 16 bit GOT entry */
#define R_M32R_GOTPC_HI_ULO  59    /* High 16 bit PC relative offset to
                                GOT with unsigned low */
#define R_M32R_GOTPC_HI_SLO  60    /* High 16 bit PC relative offset to
                                GOT with signed low */
#define R_M32R_GOTPC_LO         61    /* Low 16 bit PC relative offset to
                                GOT */
#define R_M32R_GOTOFF_HI_ULO       62    /* High 16 bit offset to GOT
                                with unsigned low */
#define R_M32R_GOTOFF_HI_SLO63    /* High 16 bit offset to GOT
                                with signed low */
#define R_M32R_GOTOFF_LO       64    /* Low 16 bit offset to GOT */
#define R_M32R_NUM          256  /* Keep this the last entry. */

/* MicroBlaze relocations */
#define R_MICROBLAZE_NONE           0     /* No reloc. */
#define R_MICROBLAZE_32         1     /* Direct 32 bit. */
#define R_MICROBLAZE_32_PCREL           2     /* PC relative 32 bit. */
#define R_MICROBLAZE_64_PCREL           3     /* PC relative 64 bit. */
#define R_MICROBLAZE_32_PCREL_LO 4     /* Low 16 bits of PCREL32. */
#define R_MICROBLAZE_64         5     /* Direct 64 bit. */
#define R_MICROBLAZE_32_LO          6     /* Low 16 bit. */
#define R_MICROBLAZE_SRO32          7     /* Read-only small data area. */
#define R_MICROBLAZE_SRW32          8     /* Read-write small data area. */
#define R_MICROBLAZE_64_NONE            9     /* No reloc. */
#define R_MICROBLAZE_32_SYM_OP_SYM  10    /* Symbol Op Symbol relocation. */
#define R_MICROBLAZE_GNU_VTINHERIT    11    /* GNU C++ vtable hierarchy. */
```

```
#define R_MICROBLAZE_GNU_VTENTRY      12   /* GNU C++ vtable member usage. */
#define R_MICROBLAZE_GOTPC_64         13   /* PC-relative GOT offset.    */
#define R_MICROBLAZE_GOT_64       14   /* GOT entry offset.    */
#define R_MICROBLAZE_PLT_64       15   /* PLT offset (PC-relative).    */
#define R_MICROBLAZE_REL      16   /* Adjust by program base.    */
#define R_MICROBLAZE_JUMP_SLOT        17   /* Create PLT entry.    */
#define R_MICROBLAZE_GLOB_DAT         18   /* Create GOT entry.    */
#define R_MICROBLAZE_GOTOFF_64        19   /* 64 bit offset to GOT. */
#define R_MICROBLAZE_GOTOFF_32        20   /* 32 bit offset to GOT. */
#define R_MICROBLAZE_COPY         21   /* Runtime copy.    */
#define R_MICROBLAZE_TLS      22   /* TLS Reloc. */
#define R_MICROBLAZE_TLSGD        23   /* TLS General Dynamic. */
#define R_MICROBLAZE_TLSLD        24   /* TLS Local Dynamic. */
#define R_MICROBLAZE_TLSDTPMOD32      25   /* TLS Module ID. */
#define R_MICROBLAZE_TLSDTPREL32 26   /* TLS Offset Within TLS Block. */
#define R_MICROBLAZE_TLSDTPREL64 27   /* TLS Offset Within TLS Block. */
#define R_MICROBLAZE_TLSGOTTPREL32    28   /* TLS Offset From Thread Pointer. */
#define R_MICROBLAZE_TLSTPREL32   29   /* TLS Offset From Thread Pointer. */

/* Legal values for d_tag (dynamic entry type).    */
#define DT_NIOS2_GP                   0x70000002 /* Address of _gp.    */

/* Nios II relocations.    */
#define R_NIOS2_NONE      0     /* No reloc.    */
#define R_NIOS2_S16       1     /* Direct signed 16 bit.    */
#define R_NIOS2_U16       2     /* Direct unsigned 16 bit.    */
#define R_NIOS2_PCREL16       3     /* PC relative 16 bit.    */
#define R_NIOS2_CALL26        4     /* Direct call.    */
#define R_NIOS2_IMM5      5     /* 5 bit constant expression.    */
#define R_NIOS2_CACHE_OPX     6     /* 5 bit expression, shift 22.    */
#define R_NIOS2_IMM6      7     /* 6 bit constant expression.    */
#define R_NIOS2_IMM8      8     /* 8 bit constant expression.    */
#define R_NIOS2_HI16      9     /* High 16 bit.    */
#define R_NIOS2_LO16      10    /* Low 16 bit.    */
#define R_NIOS2_HIADJ16       11    /* High 16 bit, adjusted.    */
#define R_NIOS2_BFD_RELOC_32 12   /* 32 bit symbol value + addend.    */
#define R_NIOS2_BFD_RELOC_16 13   /* 16 bit symbol value + addend.    */
#define R_NIOS2_BFD_RELOC_8   14   /* 8 bit symbol value + addend.    */
#define R_NIOS2_GPREL     15    /* 16 bit GP pointer offset.    */
#define R_NIOS2_GNU_VTINHERIT     16    /* GNU C++ vtable hierarchy.    */
#define R_NIOS2_GNU_VTENTRY  17   /* GNU C++ vtable member usage.    */
#define R_NIOS2_UJMP      18    /* Unconditional branch.    */
#define R_NIOS2_CJMP      19    /* Conditional branch.    */
#define R_NIOS2_CALLR     20    /* Indirect call through register.    */
```

```
#define R_NIOS2_ALIGN        21   /* Alignment requirement for
                                  linker relaxation.   */
#define R_NIOS2_GOT16        22   /* 16 bit GOT entry.   */
#define R_NIOS2_CALL16       23   /* 16 bit GOT entry for function.   */
#define R_NIOS2_GOTOFF_LO    24   /* %lo of offset to GOT pointer.   */
#define R_NIOS2_GOTOFF_HA    25   /* %hiadj of offset to GOT pointer.   */
#define R_NIOS2_PCREL_LO  26   /* %lo of PC relative offset.   */
#define R_NIOS2_PCREL_HA  27   /* %hiadj of PC relative offset.   */
#define R_NIOS2_TLS_GD16  28   /* 16 bit GOT offset for TLS GD.   */
#define R_NIOS2_TLS_LDM16    29   /* 16 bit GOT offset for TLS LDM.   */
#define R_NIOS2_TLS_LDO16 30   /* 16 bit module relative offset.   */
#define R_NIOS2_TLS_IE16   31   /* 16 bit GOT offset for TLS IE.   */
#define R_NIOS2_TLS_LE16   32   /* 16 bit LE TP-relative offset.   */
#define R_NIOS2_TLS_DTPMOD   33   /* Module number.   */
#define R_NIOS2_TLS_DTPREL   34   /* Module-relative offset.   */
#define R_NIOS2_TLS_TPREL 35   /* TP-relative offset.   */
#define R_NIOS2_COPY         36   /* Copy symbol at runtime.   */
#define R_NIOS2_GLOB_DAT 37   /* Create GOT entry.   */
#define R_NIOS2_JUMP_SLOT    38   /* Create PLT entry.   */
#define R_NIOS2_RELATIVE   39   /* Adjust by program base.   */
#define R_NIOS2_GOTOFF       40   /* 16 bit offset to GOT pointer.   */
#define R_NIOS2_CALL26_NOAT   41   /* Direct call in .noat section.   */
#define R_NIOS2_GOT_LO       42   /* %lo() of GOT entry.   */
#define R_NIOS2_GOT_HA       43   /* %hiadj() of GOT entry.   */
#define R_NIOS2_CALL_LO      44   /* %lo() of function GOT entry.   */
#define R_NIOS2_CALL_HA      45   /* %hiadj() of function GOT entry.   */

/* TILEPro relocations.   */
#define R_TILEPRO_NONE       0     /* No reloc */
#define R_TILEPRO_32         1     /* Direct 32 bit */
#define R_TILEPRO_16         2     /* Direct 16 bit */
#define R_TILEPRO_8          3     /* Direct 8 bit */
#define R_TILEPRO_32_PCREL   4     /* PC relative 32 bit */
#define R_TILEPRO_16_PCREL   5     /* PC relative 16 bit */
#define R_TILEPRO_8_PCREL 6     /* PC relative 8 bit */
#define R_TILEPRO_LO16       7     /* Low 16 bit */
#define R_TILEPRO_HI16       8     /* High 16 bit */
#define R_TILEPRO_HA16       9     /* High 16 bit, adjusted */
#define R_TILEPRO_COPY       10    /* Copy relocation */
#define R_TILEPRO_GLOB_DAT   11    /* Create GOT entry */
#define R_TILEPRO_JMP_SLOT   12    /* Create PLT entry */
#define R_TILEPRO_RELATIVE13    /* Adjust by program base */
#define R_TILEPRO_BROFF_X1   14    /* X1 pipe branch offset */
#define R_TILEPRO_JOFFLONG_X115    /* X1 pipe jump offset */
```

```
#define R_TILEPRO_JOFFLONG_X1_PLT 16    /* X1 pipe jump offset to PLT */
#define R_TILEPRO_IMM8_X0       17    /* X0 pipe 8-bit */
#define R_TILEPRO_IMM8_Y0       18    /* Y0 pipe 8-bit */
#define R_TILEPRO_IMM8_X1       19    /* X1 pipe 8-bit */
#define R_TILEPRO_IMM8_Y1       20    /* Y1 pipe 8-bit */
#define R_TILEPRO_MT_IMM15_X1      21    /* X1 pipe mtspr */
#define R_TILEPRO_MF_IMM15_X1      22    /* X1 pipe mfspr */
#define R_TILEPRO_IMM16_X0      23    /* X0 pipe 16-bit */
#define R_TILEPRO_IMM16_X1      24    /* X1 pipe 16-bit */
#define R_TILEPRO_IMM16_X0_LO       25    /* X0 pipe low 16-bit */
#define R_TILEPRO_IMM16_X1_LO       26    /* X1 pipe low 16-bit */
#define R_TILEPRO_IMM16_X0_HI       27    /* X0 pipe high 16-bit */
#define R_TILEPRO_IMM16_X1_HI       28    /* X1 pipe high 16-bit */
#define R_TILEPRO_IMM16_X0_HA       29    /* X0 pipe high 16-bit, adjusted */
#define R_TILEPRO_IMM16_X1_HA       30    /* X1 pipe high 16-bit, adjusted */
#define R_TILEPRO_IMM16_X0_PCREL 31    /* X0 pipe PC relative 16 bit */
#define R_TILEPRO_IMM16_X1_PCREL 32    /* X1 pipe PC relative 16 bit */
#define R_TILEPRO_IMM16_X0_LO_PCREL 33      /* X0 pipe PC relative low 16 bit */
#define R_TILEPRO_IMM16_X1_LO_PCREL 34      /* X1 pipe PC relative low 16 bit */
#define R_TILEPRO_IMM16_X0_HI_PCREL 35      /* X0 pipe PC relative high 16 bit */
#define R_TILEPRO_IMM16_X1_HI_PCREL 36      /* X1 pipe PC relative high 16 bit */
#define R_TILEPRO_IMM16_X0_HA_PCREL 37      /* X0 pipe PC relative ha() 16 bit */
#define R_TILEPRO_IMM16_X1_HA_PCREL 38      /* X1 pipe PC relative ha() 16 bit */
#define R_TILEPRO_IMM16_X0_GOT    39    /* X0 pipe 16-bit GOT offset */
#define R_TILEPRO_IMM16_X1_GOT    40    /* X1 pipe 16-bit GOT offset */
#define R_TILEPRO_IMM16_X0_GOT_LO 41 /* X0 pipe low 16-bit GOT offset */
#define R_TILEPRO_IMM16_X1_GOT_LO 42 /* X1 pipe low 16-bit GOT offset */
#define R_TILEPRO_IMM16_X0_GOT_HI 43  /* X0 pipe high 16-bit GOT offset */
#define R_TILEPRO_IMM16_X1_GOT_HI 44  /* X1 pipe high 16-bit GOT offset */
#define R_TILEPRO_IMM16_X0_GOT_HA 45 /* X0 pipe ha() 16-bit GOT offset */
#define R_TILEPRO_IMM16_X1_GOT_HA 46 /* X1 pipe ha() 16-bit GOT offset */
#define R_TILEPRO_MMSTART_X0 47    /* X0 pipe mm "start" */
#define R_TILEPRO_MMEND_X0    48    /* X0 pipe mm "end" */
#define R_TILEPRO_MMSTART_X1 49    /* X1 pipe mm "start" */
#define R_TILEPRO_MMEND_X1    50    /* X1 pipe mm "end" */
#define R_TILEPRO_SHAMT_X0      51    /* X0 pipe shift amount */
#define R_TILEPRO_SHAMT_X1      52    /* X1 pipe shift amount */
#define R_TILEPRO_SHAMT_Y0      53    /* Y0 pipe shift amount */
#define R_TILEPRO_SHAMT_Y1      54    /* Y1 pipe shift amount */
#define R_TILEPRO_DEST_IMM8_X1      55    /* X1 pipe destination 8-bit */
/* Relocs 56-59 are currently not defined.    */
#define R_TILEPRO_TLS_GD_CALL 60    /* "jal" for TLS GD */
#define R_TILEPRO_IMM8_X0_TLS_GD_ADD 61   /* X0 pipe "addi" for TLS GD */
#define R_TILEPRO_IMM8_X1_TLS_GD_ADD 62   /* X1 pipe "addi" for TLS GD */
```

```
#define R_TILEPRO_IMM8_Y0_TLS_GD_ADD 63   /* Y0 pipe "addi" for TLS GD */
#define R_TILEPRO_IMM8_Y1_TLS_GD_ADD 64   /* Y1 pipe "addi" for TLS GD */
#define R_TILEPRO_TLS_IE_LOAD  65    /* "lw_tls" for TLS IE */
#define R_TILEPRO_IMM16_X0_TLS_GD 66  /* X0 pipe 16-bit TLS GD offset */
#define R_TILEPRO_IMM16_X1_TLS_GD 67  /* X1 pipe 16-bit TLS GD offset */
#define R_TILEPRO_IMM16_X0_TLS_GD_LO 68    /* X0 pipe low 16-bit TLS GD offset */
#define R_TILEPRO_IMM16_X1_TLS_GD_LO 69    /* X1 pipe low 16-bit TLS GD offset */
#define R_TILEPRO_IMM16_X0_TLS_GD_HI 70    /* X0 pipe high 16-bit TLS GD offset */
#define R_TILEPRO_IMM16_X1_TLS_GD_HI 71    /* X1 pipe high 16-bit TLS GD offset */
#define R_TILEPRO_IMM16_X0_TLS_GD_HA 72   /* X0 pipe ha() 16-bit TLS GD offset */
#define R_TILEPRO_IMM16_X1_TLS_GD_HA 73   /* X1 pipe ha() 16-bit TLS GD offset */
#define R_TILEPRO_IMM16_X0_TLS_IE 74    /* X0 pipe 16-bit TLS IE offset */
#define R_TILEPRO_IMM16_X1_TLS_IE 75    /* X1 pipe 16-bit TLS IE offset */
#define R_TILEPRO_IMM16_X0_TLS_IE_LO 76     /* X0 pipe low 16-bit TLS IE offset */
#define R_TILEPRO_IMM16_X1_TLS_IE_LO 77     /* X1 pipe low 16-bit TLS IE offset */
#define R_TILEPRO_IMM16_X0_TLS_IE_HI 78     /* X0 pipe high 16-bit TLS IE offset */
#define R_TILEPRO_IMM16_X1_TLS_IE_HI 79     /* X1 pipe high 16-bit TLS IE offset */
#define R_TILEPRO_IMM16_X0_TLS_IE_HA 80     /* X0 pipe ha() 16-bit TLS IE offset */
#define R_TILEPRO_IMM16_X1_TLS_IE_HA 81     /* X1 pipe ha() 16-bit TLS IE offset */
#define R_TILEPRO_TLS_DTPMOD32     82    /* ID of module containing symbol */
#define R_TILEPRO_TLS_DTPOFF32     83    /* Offset in TLS block */
#define R_TILEPRO_TLS_TPOFF32 84    /* Offset in static TLS block */
#define R_TILEPRO_IMM16_X0_TLS_LE 85   /* X0 pipe 16-bit TLS LE offset */
#define R_TILEPRO_IMM16_X1_TLS_LE 86   /* X1 pipe 16-bit TLS LE offset */
#define R_TILEPRO_IMM16_X0_TLS_LE_LO 87     /* X0 pipe low 16-bit TLS LE offset */
#define R_TILEPRO_IMM16_X1_TLS_LE_LO 88     /* X1 pipe low 16-bit TLS LE offset */
#define R_TILEPRO_IMM16_X0_TLS_LE_HI 89     /* X0 pipe high 16-bit TLS LE offset */
#define R_TILEPRO_IMM16_X1_TLS_LE_HI 90     /* X1 pipe high 16-bit TLS LE offset */
#define R_TILEPRO_IMM16_X0_TLS_LE_HA 91    /* X0 pipe ha() 16-bit TLS LE offset */
#define R_TILEPRO_IMM16_X1_TLS_LE_HA 92    /* X1 pipe ha() 16-bit TLS LE offset */

#define R_TILEPRO_GNU_VTINHERIT    128  /* GNU C++ vtable hierarchy */
#define R_TILEPRO_GNU_VTENTRY      129  /* GNU C++ vtable member usage */

#define R_TILEPRO_NUM              130


/* TILE-Gx relocations.    */
#define R_TILEGX_NONE              0     /* No reloc */
#define R_TILEGX_64                1     /* Direct 64 bit */
#define R_TILEGX_32                2     /* Direct 32 bit */
#define R_TILEGX_16                3     /* Direct 16 bit */
#define R_TILEGX_8                 4     /* Direct 8 bit */
#define R_TILEGX_64_PCREL 5     /* PC relative 64 bit */
```

```
#define R_TILEGX_32_PCREL 6     /* PC relative 32 bit */
#define R_TILEGX_16_PCREL 7     /* PC relative 16 bit */
#define R_TILEGX_8_PCREL  8     /* PC relative 8 bit */
#define R_TILEGX_HW0      9     /* hword 0 16-bit */
#define R_TILEGX_HW1      10    /* hword 1 16-bit */
#define R_TILEGX_HW2      11    /* hword 2 16-bit */
#define R_TILEGX_HW3      12    /* hword 3 16-bit */
#define R_TILEGX_HW0_LAST      13   /* last hword 0 16-bit */
#define R_TILEGX_HW1_LAST      14   /* last hword 1 16-bit */
#define R_TILEGX_HW2_LAST      15   /* last hword 2 16-bit */
#define R_TILEGX_COPY      16   /* Copy relocation */
#define R_TILEGX_GLOB_DAT17   /* Create GOT entry */
#define R_TILEGX_JMP_SLOT 18   /* Create PLT entry */
#define R_TILEGX_RELATIVE  19   /* Adjust by program base */
#define R_TILEGX_BROFF_X1 20   /* X1 pipe branch offset */
#define R_TILEGX_JUMPOFF_X1    21   /* X1 pipe jump offset */
#define R_TILEGX_JUMPOFF_X1_PLT    22   /* X1 pipe jump offset to PLT */
#define R_TILEGX_IMM8_X0  23   /* X0 pipe 8-bit */
#define R_TILEGX_IMM8_Y0  24   /* Y0 pipe 8-bit */
#define R_TILEGX_IMM8_X1  25   /* X1 pipe 8-bit */
#define R_TILEGX_IMM8_Y1  26   /* Y1 pipe 8-bit */
#define R_TILEGX_DEST_IMM8_X1      27   /* X1 pipe destination 8-bit */
#define R_TILEGX_MT_IMM14_X128   /* X1 pipe mtspr */
#define R_TILEGX_MF_IMM14_X129   /* X1 pipe mfspr */
#define R_TILEGX_MMSTART_X0  30   /* X0 pipe mm "start" */
#define R_TILEGX_MMEND_X0     31   /* X0 pipe mm "end" */
#define R_TILEGX_SHAMT_X0      32   /* X0 pipe shift amount */
#define R_TILEGX_SHAMT_X1      33   /* X1 pipe shift amount */
#define R_TILEGX_SHAMT_Y0      34   /* Y0 pipe shift amount */
#define R_TILEGX_SHAMT_Y1      35   /* Y1 pipe shift amount */
#define R_TILEGX_IMM16_X0_HW0     36   /* X0 pipe hword 0 */
#define R_TILEGX_IMM16_X1_HW0     37   /* X1 pipe hword 0 */
#define R_TILEGX_IMM16_X0_HW1     38   /* X0 pipe hword 1 */
#define R_TILEGX_IMM16_X1_HW1     39   /* X1 pipe hword 1 */
#define R_TILEGX_IMM16_X0_HW2     40   /* X0 pipe hword 2 */
#define R_TILEGX_IMM16_X1_HW2     41   /* X1 pipe hword 2 */
#define R_TILEGX_IMM16_X0_HW3     42   /* X0 pipe hword 3 */
#define R_TILEGX_IMM16_X1_HW3     43   /* X1 pipe hword 3 */
#define R_TILEGX_IMM16_X0_HW0_LAST 44      /* X0 pipe last hword 0 */
#define R_TILEGX_IMM16_X1_HW0_LAST 45      /* X1 pipe last hword 0 */
#define R_TILEGX_IMM16_X0_HW1_LAST 46      /* X0 pipe last hword 1 */
#define R_TILEGX_IMM16_X1_HW1_LAST 47      /* X1 pipe last hword 1 */
#define R_TILEGX_IMM16_X0_HW2_LAST 48      /* X0 pipe last hword 2 */
#define R_TILEGX_IMM16_X1_HW2_LAST 49      /* X1 pipe last hword 2 */
```

```
#define R_TILEGX_IMM16_X0_HW0_PCREL 50     /* X0 pipe PC relative hword 0 */
#define R_TILEGX_IMM16_X1_HW0_PCREL 51     /* X1 pipe PC relative hword 0 */
#define R_TILEGX_IMM16_X0_HW1_PCREL 52     /* X0 pipe PC relative hword 1 */
#define R_TILEGX_IMM16_X1_HW1_PCREL 53     /* X1 pipe PC relative hword 1 */
#define R_TILEGX_IMM16_X0_HW2_PCREL 54     /* X0 pipe PC relative hword 2 */
#define R_TILEGX_IMM16_X1_HW2_PCREL 55     /* X1 pipe PC relative hword 2 */
#define R_TILEGX_IMM16_X0_HW3_PCREL 56     /* X0 pipe PC relative hword 3 */
#define R_TILEGX_IMM16_X1_HW3_PCREL 57     /* X1 pipe PC relative hword 3 */
#define R_TILEGX_IMM16_X0_HW0_LAST_PCREL 58 /* X0 pipe PC-rel last hword 0 */
#define R_TILEGX_IMM16_X1_HW0_LAST_PCREL 59 /* X1 pipe PC-rel last hword 0 */
#define R_TILEGX_IMM16_X0_HW1_LAST_PCREL 60 /* X0 pipe PC-rel last hword 1 */
#define R_TILEGX_IMM16_X1_HW1_LAST_PCREL 61 /* X1 pipe PC-rel last hword 1 */
#define R_TILEGX_IMM16_X0_HW2_LAST_PCREL 62 /* X0 pipe PC-rel last hword 2 */
#define R_TILEGX_IMM16_X1_HW2_LAST_PCREL 63 /* X1 pipe PC-rel last hword 2 */
#define R_TILEGX_IMM16_X0_HW0_GOT 64       /* X0 pipe hword 0 GOT offset */
#define R_TILEGX_IMM16_X1_HW0_GOT 65       /* X1 pipe hword 0 GOT offset */
#define R_TILEGX_IMM16_X0_HW0_PLT_PCREL 66 /* X0 pipe PC-rel PLT hword 0 */
#define R_TILEGX_IMM16_X1_HW0_PLT_PCREL 67 /* X1 pipe PC-rel PLT hword 0 */
#define R_TILEGX_IMM16_X0_HW1_PLT_PCREL 68 /* X0 pipe PC-rel PLT hword 1 */
#define R_TILEGX_IMM16_X1_HW1_PLT_PCREL 69 /* X1 pipe PC-rel PLT hword 1 */
#define R_TILEGX_IMM16_X0_HW2_PLT_PCREL 70 /* X0 pipe PC-rel PLT hword 2 */
#define R_TILEGX_IMM16_X1_HW2_PLT_PCREL 71 /* X1 pipe PC-rel PLT hword 2 */
#define R_TILEGX_IMM16_X0_HW0_LAST_GOT 72 /* X0 pipe last hword 0 GOT offset */
#define R_TILEGX_IMM16_X1_HW0_LAST_GOT 73 /* X1 pipe last hword 0 GOT offset */
#define R_TILEGX_IMM16_X0_HW1_LAST_GOT 74 /* X0 pipe last hword 1 GOT offset */
#define R_TILEGX_IMM16_X1_HW1_LAST_GOT 75 /* X1 pipe last hword 1 GOT offset */
#define R_TILEGX_IMM16_X0_HW3_PLT_PCREL 76 /* X0 pipe PC-rel PLT hword 3 */
#define R_TILEGX_IMM16_X1_HW3_PLT_PCREL 77 /* X1 pipe PC-rel PLT hword 3 */
#define R_TILEGX_IMM16_X0_HW0_TLS_GD 78  /* X0 pipe hword 0 TLS GD offset */
#define R_TILEGX_IMM16_X1_HW0_TLS_GD 79  /* X1 pipe hword 0 TLS GD offset */
#define R_TILEGX_IMM16_X0_HW0_TLS_LE 80    /* X0 pipe hword 0 TLS LE offset */
#define R_TILEGX_IMM16_X1_HW0_TLS_LE 81    /* X1 pipe hword 0 TLS LE offset */
#define R_TILEGX_IMM16_X0_HW0_LAST_TLS_LE 82 /* X0 pipe last hword 0 LE off */
#define R_TILEGX_IMM16_X1_HW0_LAST_TLS_LE 83 /* X1 pipe last hword 0 LE off */
#define R_TILEGX_IMM16_X0_HW1_LAST_TLS_LE 84 /* X0 pipe last hword 1 LE off */
#define R_TILEGX_IMM16_X1_HW1_LAST_TLS_LE 85 /* X1 pipe last hword 1 LE off */
#define R_TILEGX_IMM16_X0_HW0_LAST_TLS_GD 86 /* X0 pipe last hword 0 GD off */
#define R_TILEGX_IMM16_X1_HW0_LAST_TLS_GD 87 /* X1 pipe last hword 0 GD off */
#define R_TILEGX_IMM16_X0_HW1_LAST_TLS_GD 88 /* X0 pipe last hword 1 GD off */
#define R_TILEGX_IMM16_X1_HW1_LAST_TLS_GD 89 /* X1 pipe last hword 1 GD off */
/* Relocs 90-91 are currently not defined.    */
#define R_TILEGX_IMM16_X0_HW0_TLS_IE 92    /* X0 pipe hword 0 TLS IE offset */
#define R_TILEGX_IMM16_X1_HW0_TLS_IE 93    /* X1 pipe hword 0 TLS IE offset */
#define R_TILEGX_IMM16_X0_HW0_LAST_PLT_PCREL 94 /* X0 pipe PC-rel PLT last hword 0 */
```

```
#define R_TILEGX_IMM16_X1_HW0_LAST_PLT_PCREL 95 /* X1 pipe PC-rel PLT last hword 0 */
#define R_TILEGX_IMM16_X0_HW1_LAST_PLT_PCREL 96 /* X0 pipe PC-rel PLT last hword 1 */
#define R_TILEGX_IMM16_X1_HW1_LAST_PLT_PCREL 97 /* X1 pipe PC-rel PLT last hword 1 */
#define R_TILEGX_IMM16_X0_HW2_LAST_PLT_PCREL 98 /* X0 pipe PC-rel PLT last hword 2 */
#define R_TILEGX_IMM16_X1_HW2_LAST_PLT_PCREL 99 /* X1 pipe PC-rel PLT last hword 2 */
#define R_TILEGX_IMM16_X0_HW0_LAST_TLS_IE 100 /* X0 pipe last hword 0 IE off */
#define R_TILEGX_IMM16_X1_HW0_LAST_TLS_IE 101 /* X1 pipe last hword 0 IE off */
#define R_TILEGX_IMM16_X0_HW1_LAST_TLS_IE 102 /* X0 pipe last hword 1 IE off */
#define R_TILEGX_IMM16_X1_HW1_LAST_TLS_IE 103 /* X1 pipe last hword 1 IE off */
/* Relocs 104-105 are currently not defined.    */
#define R_TILEGX_TLS_DTPMOD64       106  /* 64-bit ID of symbol's module */
#define R_TILEGX_TLS_DTPOFF64 107  /* 64-bit offset in TLS block */
#define R_TILEGX_TLS_TPOFF64    108  /* 64-bit offset in static TLS block */
#define R_TILEGX_TLS_DTPMOD32       109  /* 32-bit ID of symbol's module */
#define R_TILEGX_TLS_DTPOFF32 110  /* 32-bit offset in TLS block */
#define R_TILEGX_TLS_TPOFF32    111  /* 32-bit offset in static TLS block */
#define R_TILEGX_TLS_GD_CALL    112  /* "jal" for TLS GD */
#define R_TILEGX_IMM8_X0_TLS_GD_ADD 113   /* X0 pipe "addi" for TLS GD */
#define R_TILEGX_IMM8_X1_TLS_GD_ADD 114   /* X1 pipe "addi" for TLS GD */
#define R_TILEGX_IMM8_Y0_TLS_GD_ADD 115   /* Y0 pipe "addi" for TLS GD */
#define R_TILEGX_IMM8_Y1_TLS_GD_ADD 116   /* Y1 pipe "addi" for TLS GD */
#define R_TILEGX_TLS_IE_LOAD    117  /* "ld_tls" for TLS IE */
#define R_TILEGX_IMM8_X0_TLS_ADD 118  /* X0 pipe "addi" for TLS GD/IE */
#define R_TILEGX_IMM8_X1_TLS_ADD 119  /* X1 pipe "addi" for TLS GD/IE */
#define R_TILEGX_IMM8_Y0_TLS_ADD 120  /* Y0 pipe "addi" for TLS GD/IE */
#define R_TILEGX_IMM8_Y1_TLS_ADD 121  /* Y1 pipe "addi" for TLS GD/IE */

#define R_TILEGX_GNU_VTINHERIT      128  /* GNU C++ vtable hierarchy */
#define R_TILEGX_GNU_VTENTRY 129  /* GNU C++ vtable member usage */

#define R_TILEGX_NUM        130


__END_DECLS

#endif     /* elf.h */
```