Automatic Porting of Binary File Descriptor Library

Maghsoud Abbaspour

Electrical and Computer Engineering
University of Toronto

Oct 19, 2001

maghsoud@eecg.toronto.edu

Outline

- Motivation.
- Retargetability(What, Why).
- Architecture Description Languages.
- Retargetable binary File Descriptor library.
- Binary File Format(ELF).
- Application Binary Interface(ABI) and ABI modeling.
- Relocation(Definition and modeling).
- PLT and GOT (definitions and PLT modeling).
- Implementation.
- Result.

Retargetability, what?

A software is said to be retargetable if it can be applied to range of target processors. Different level for switching to another target:

- **Portability**: rewriting some part of the software to use in new target
- **Retargetability**: target characteristic is captured using target description and target dependent parts are generated automatically.
- **Machine independence** : no change.

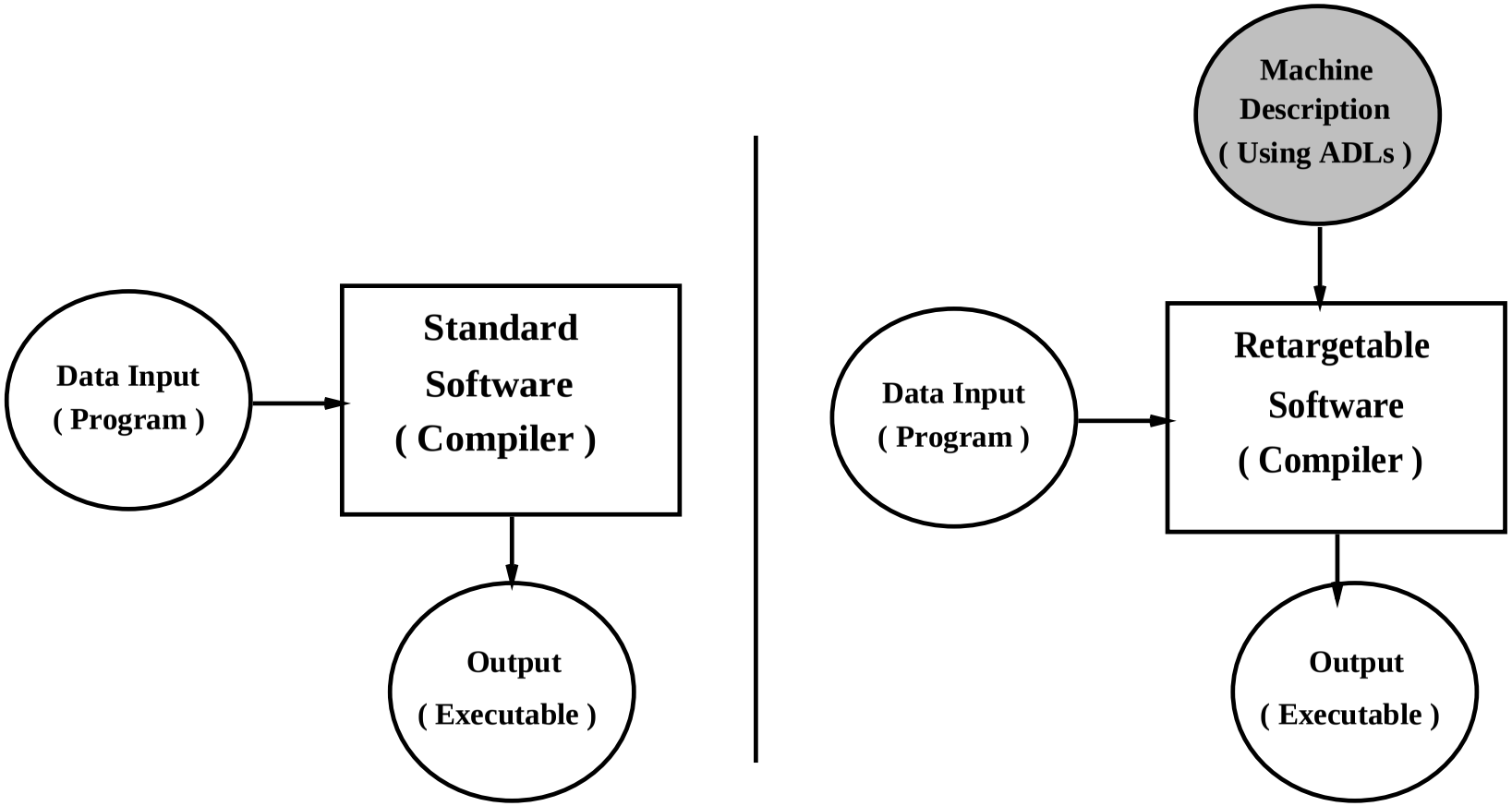
Retargetability, why?

- Time-to-Market or Time-to-Money.
- Software toolkit for new processors(ASIPs).
- Design Complexity(Expoenetially).
- Growing number of applications(Network processors, Communications processors,...).
- Small life time, high risk market, Lower-design volume

Design Automation Tool Supports

- Hardware Design Support
  - RTL model, RTL Floorplanning, Logic synthesis and ...
  - Retagrgetable Compiler and Retagrgetable Simulator for architecture exploration.
- Software Design Supoort
  - Retagrgetable Compiler.
  - Retagrgetable Simulator.
  - RetagrgetableBinary Tools(Assembler, Linker, Debugger,...).

A Retargetable Software



Architecture Description Languages(ADLs)

- HDLs can not be used to generate toolkit.
- Instruction Set Architecture(ISA) description languages.
- Architecture Exploration
  - Can not run an Application program on a HDL simulator in behavioral level.
  - Running an Application program is very time consuming on a HDL simulator in structural level.
- Simple and general(text only, simple and wide range architecture support).
- *Suport automatic tool-kit generation*(Compiler, Simulater,...)

ADLs Cont.

- Architecture View
  - micro-architecture compnents and stores information is used for Synthesis
- ISA View
  - Instruction set modeling and binary encoding information is used for Compiliation tools generation.
- Simulation View
  - Timing information is used for Simulation tools generation.

ISA Modeling

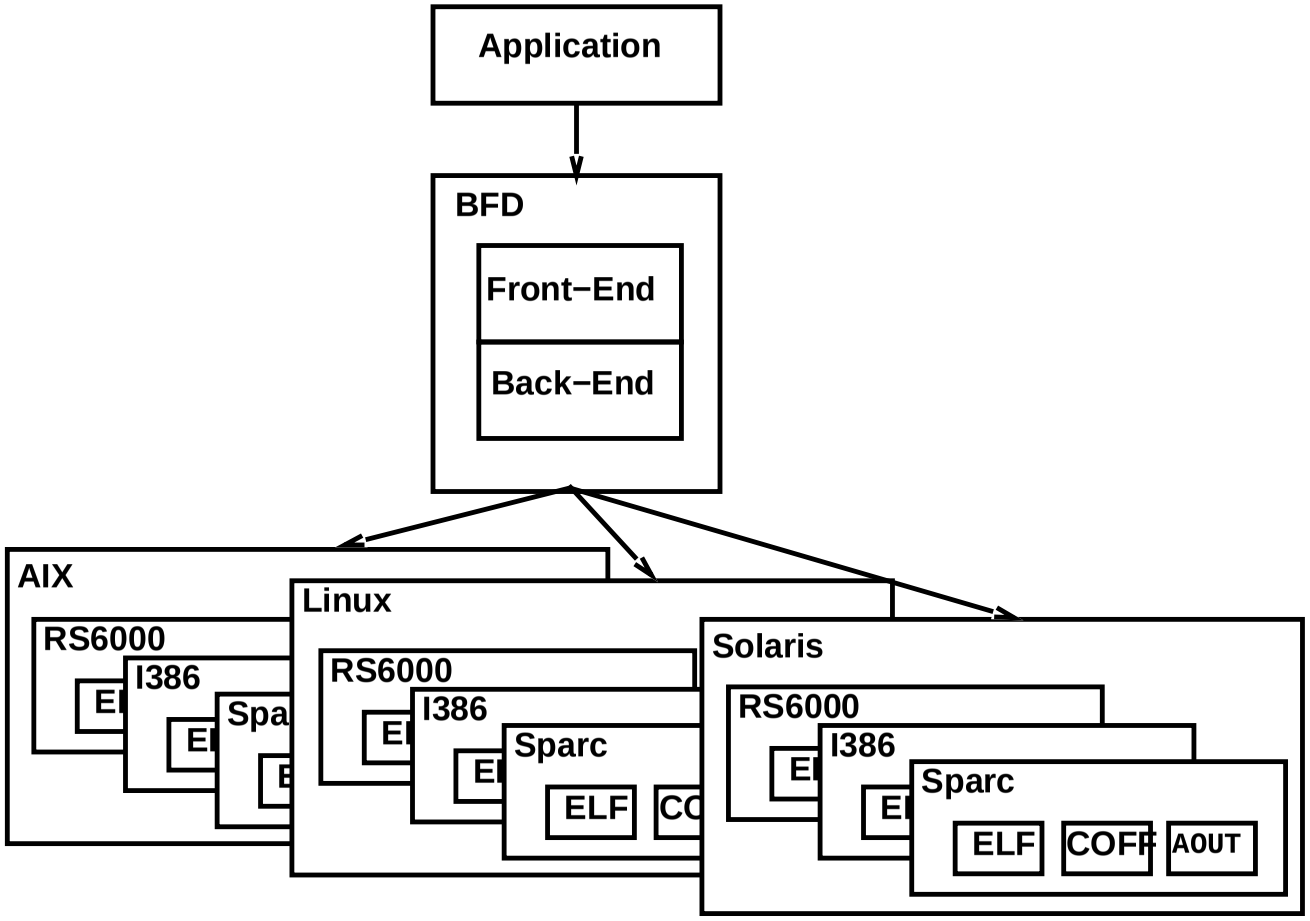
An **ISA architectural model** is a member of

$$
\begin{array}{l}
\text{ISA} = \textbf{tuple}\ \{ \\
S \qquad\qquad\qquad : \langle\rangle^{Store}; \\
I \qquad\qquad\qquad : \langle\rangle^{Instrn}; \\
P \qquad\qquad\qquad : ILP; \\
A \qquad\qquad\qquad : ABI; \\
\}
\end{array}
$$

where S is a set of **stores**, I is the set of **instructions**, P is the **instruction level parallelism and** A **is a ABI model of the micro-architecture.**

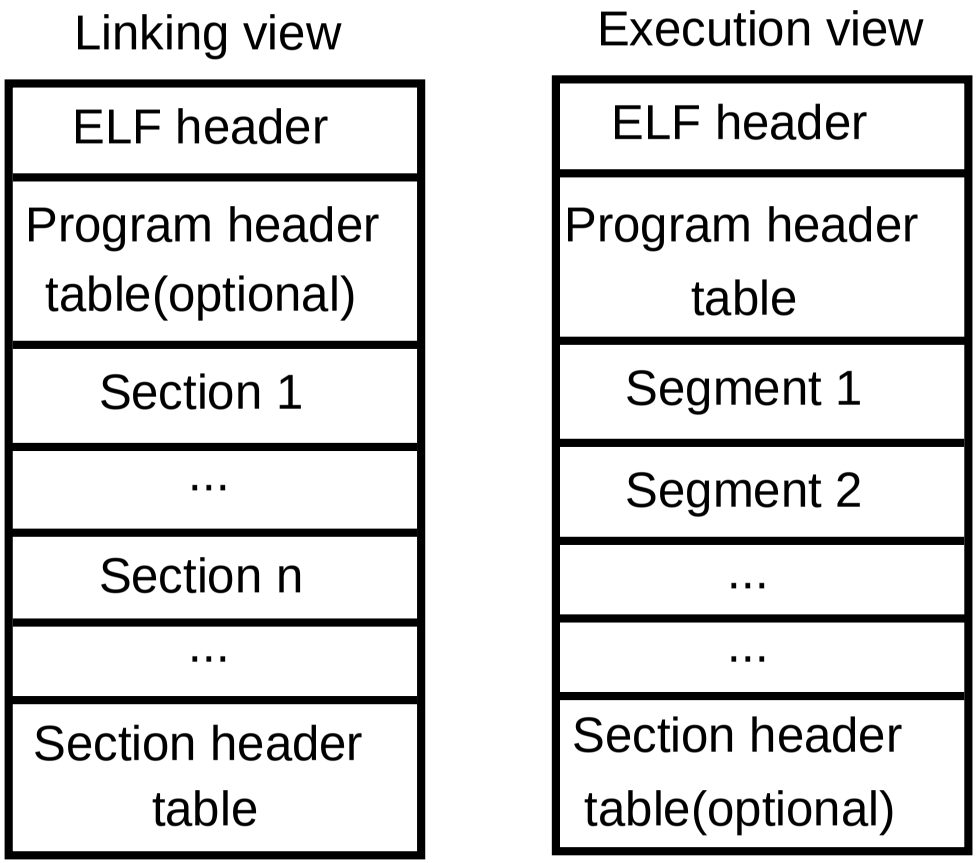
Case Study: Retagetable Binary File Descriptor(BFD) Library

What is BFD?



ELF : Binary File Format(BFF)

What is ELF?



Application Binary Interface(ABI)

What is the ABI?

- An ABI defines a binary interface for application programs that are compiled and packaged for a *specific OS* running on a *specific hardware architecture*.

ABI Component:

- Calling Convention.
- Relocation.
- PLT entries(ELF obj format specific).

Relocation

- what is relocation
- Relocations of Sparc:

Name	Calculation
R_SPARC_16	$S + A$
R_SPARC_32	$S + A$
R_SPARC_DISP32	$S + A - P$
$R\_SPARC\_WDISP30$	$S + A - P >> 2$
R_SPARC_GOT22	$G >> 10$
R_SPARC_PC22	$S + A - P >> 10$
.....	

Relocation Cont.

CALL :

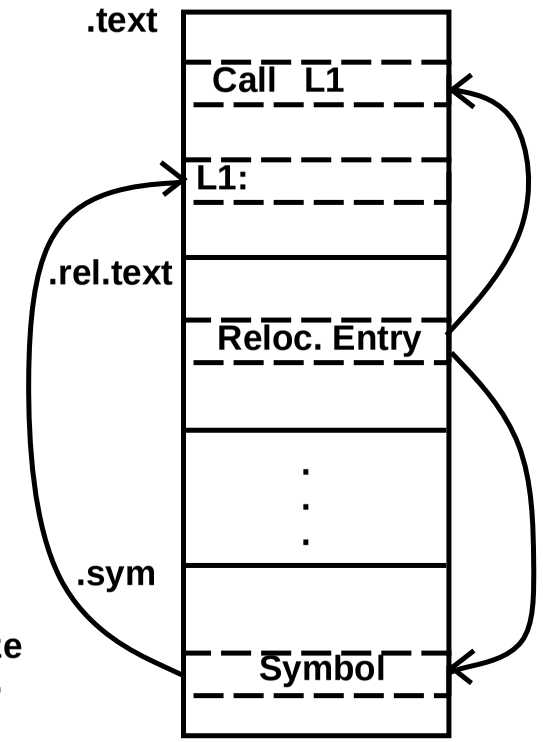
0 1	disp30

Control Transfer to "PC + (4 x disp30)

Binary Code =
0100,0000,0000,0000
and a relocation entry
will be added for this
symbol

Reloc Expr. =
S + A − P >> 2

− r_offset(P) , index (sym),
− type , addend(A)



− value(S) ,size
− name, type,...

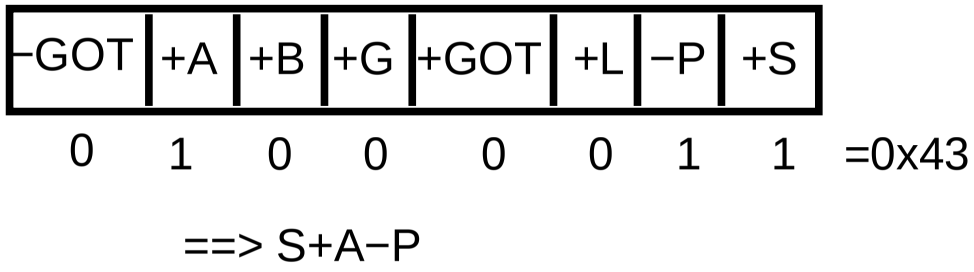
Relocation Cont.

- Relocation model A **relocation type** is a member of

$$
\begin{array}{l}
\text{Reloc} = \textbf{tuple}\ \{ \\
id \quad : int; \\
expCode \quad : byte; \\
rightshift \quad : int; \\
bitsize \quad : int; \\
bitpos \quad : int; \\
complain \quad : \\
\{ignore, bit, sign, unsigned\}; \\
\}
\end{array}
$$

Relocation Cont.

- id is an unique integer identifier.
- $expCode = \langle C_7, C_6, ..., C_0 \rangle$ encodes the expression $\Sigma_i C_i P_i$, with $P_7, ...P_0$ being $-GOT, A, B, G, GOT, L, -P, S$ respectively.

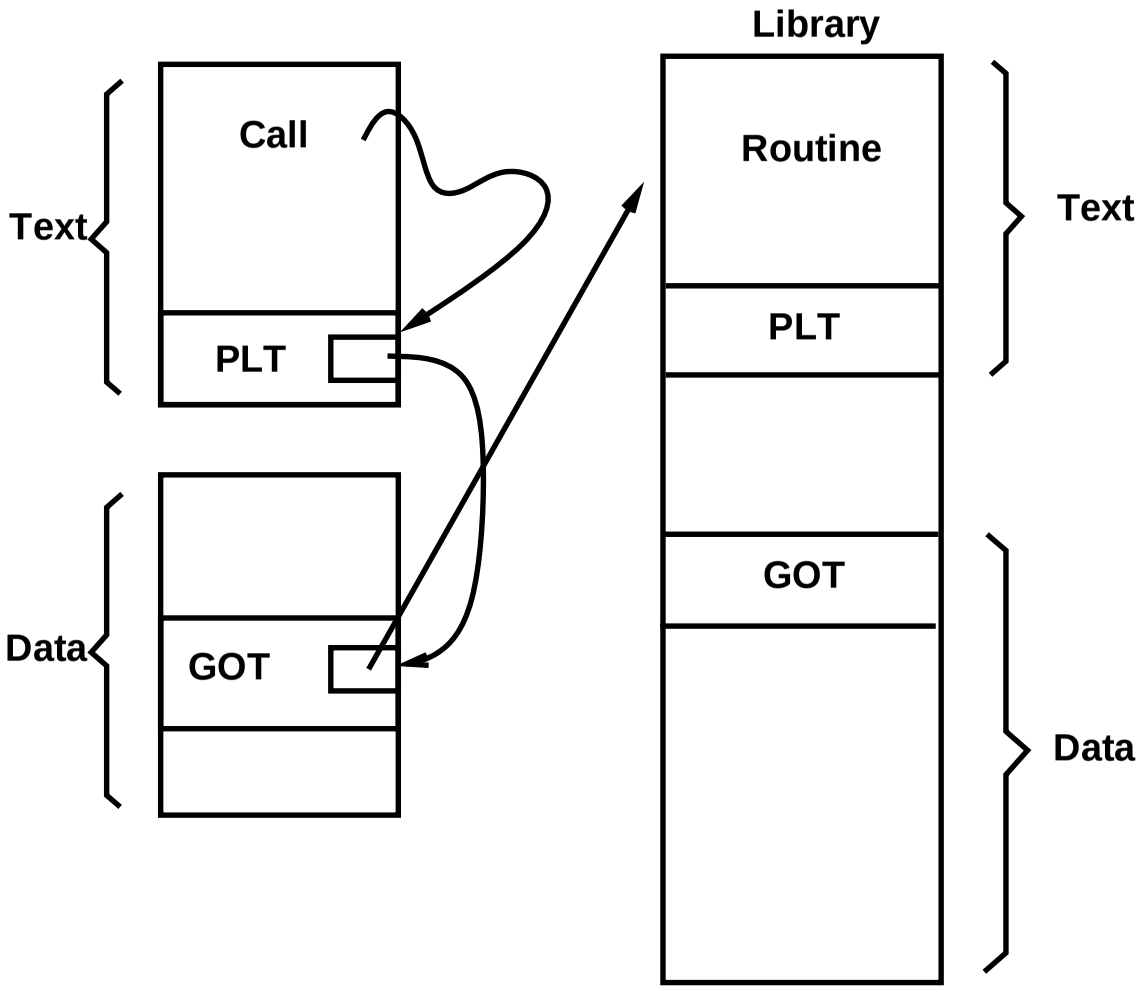


Relocation Cont.

- $rightshift$ represents the number of bits at the right side of the calculated $\Sigma_i C_i P_i$ that should be dropped.
- $bitpos$ and $bitsize$ represents the bit position as well as the size of the relocation field within the instruction or datum to be relocated.
- $complain$ encodes the action to take when specific type of overflow occurs.

PLT and GOT

Procedure Linkage Table(PLT) and Global Offset Table(GOT)



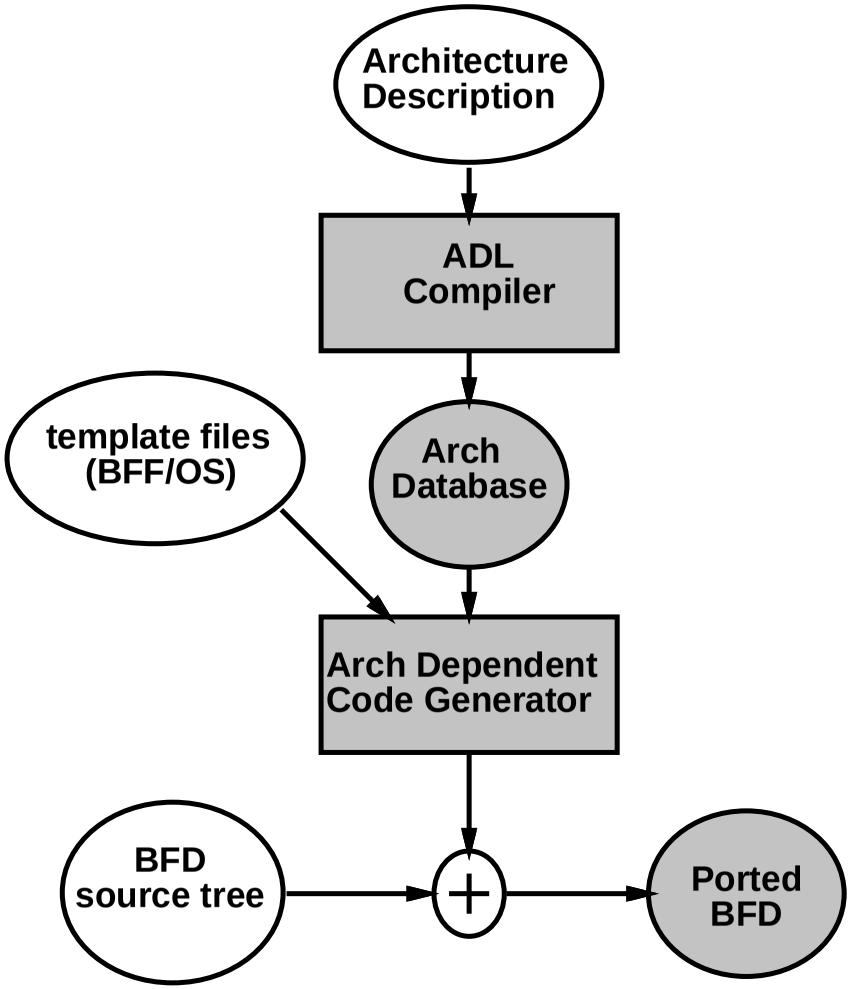
PLT modeling

A PLT entry is a member of

plt = **tuple** {
$size$: int;
$instrns$: $[\,]^{int}$;
}

- $size$ is the number of words (4 byte) for each PLT entry.
- $instrns$ is a sequence of binary words.

Implementation



Implementation

- Complexity of ABI specification.

Processor	ABI complexity (#line)
sparc	26
i386	13

- Input file Templates.
- Genrated files.

results

Generated Files	#line (generated)
/bfd/config/mysparc-elf.mt	3
/bfd/archures.c	1483
/bfd/configure.host	112
/bfd/configure.in	286
/bfd/config.bfd	166
/bfd/elf32-mysparc.c	1482
/include/elf/common.h	229
config.sub	1014
/bfd/target.c	785
/bfd/cpu-mysparc.c	44
/bfd/elfcode.h	6582
/ld/configure.in	183
/ld/emulparam/ elf32_mysparc.sh	9
/ld/Makefile.in	868
/ld/config/mysparc-elf.mt	1