

! ! " " ELF psABI # \$

! ! # \$%%&&' () *

Version 1.00

u |

+ , - . / ' O	1
> ? . + , - Q(2RS	1
C N • ; ² ^ ³ # ´	3
ELF u ` ¿ %	4
EI_CLASS: ELF ¿ % ´	4
e_machi ne: Ç È f " ID	4
e_fl ags: ABI ^ ³ É • < ` g	4
Ô Ö Æ ^ ³	6
j 2RÝ - ĭ Ÿ	10

+ , - . / ' O

Table 1.1 / + , - . / ' O

23	(2	/ 4	5) / # 6789
\$r0	\$zero	: ; 0	<: ; =
\$r1	\$ra	>? @A	7
\$r2	\$tp	* BC+	< DE F G=
\$r3	\$sp	, C+	6
\$r4-\$r5	\$a0-\$a1	- H+, - I >?. +, -	7
\$r6-\$r11	\$a2-\$a7	- H+, -	7
\$r12-\$r20	\$t0-\$t8	/ O+, -	7
\$r21		89	< DE F G=
\$r22	\$fp/\$s9	, 1 C+ / J 2+, -	6
\$r23-\$r31	\$s0-\$s8	J 2+, -	6

Table 2. KL + , - . / ' O

23	(2	/ 4	5) / # 6789
\$f0-\$f1	\$fa0-\$fa1	- H+, - I >?. +, -	7
\$f2-\$f7	\$fa2-\$fa7	- H+, -	7
\$f8-\$f23	\$ft0-\$ft15	/ O+, -	7
\$f24-\$f31	\$fs0-\$fs7	J 2+, -	6

/ O+, - MN33) / O8, +, - P J 2+, - MN33N) / O8, +, - P

>? . +, - Q(2RS

5T UVWQ LoongArch 45X6#Y 7E Z [8\] ^ \$v0 \$v1 \$fv0 \$fv1 Q+, - RS_9U2` F
(a b c \$a0 \$a1 \$fa0 \$fa1 P 9U(2de6: f MIPS QFg- HI >? . +, - RSh; <QP i
c LoongArch =>j k' ? @Q>? . +, - Y9ARSI h[mnBoY phDqC. /P

i c r s t DuQ= EFGvHY I Tw+, - J x y w ABI 2` j DT O6KLQz { P | R} QM~
LoongArch 45N• QB€DO• =EP, (2P Ef„ Q LoongArch 45X6DO• . /P, (2P

Qc!!)*...†Q‡^RS%YTfŠB3_

; < # \$€• OŽ OS%Q••• < 3 NY

NOTE

1. 5• < N' ' UOF" < V" • < = 89" • Y

2. 5• < N+1 QP/SW— —~ Y

3. 5• < N+2 X™P/SQ" • P

Qc9US%Ž OQš t DuY> Wœš t Q• ž F ^ Ÿ, 5QP/SQ" • YY š ĭ Š
BW— Yŀ• < Nŀ~ o 3ϕ T £ ꝥ œ LoongArch " • Q• w¥ ĭ Z § • < P Qc [`` Wœš
t Y © D^a \ W«] . / P / SQ' → S%- ® QS%Yš t • < ⁻ ° ± D" • P / SP

C N• ; 2 ^ 3 # ´

Table 3. LP64; 2 μ3 < QO¶ _ ABI ^ 3 _ l p64d l p64f l p64s=

´ . ^ 3	1 < ` G=	Qa < ` G=
bool / _Bool	1	1
unsigned char / char	1	1
unsigned short / short	2	2
unsigned int / int	4	4
unsigned long / long	8	8
unsigned long long / long long	8	8
C+ ^ 3	8	8
float	4	4
double	8	8
long double	16	16

Table 4. ILP32; 2 μ3 < QO¶ _ ABI ^ 3 _ i l p32d i l p32f i l p32s=

´ . ^ 3	1 < ` G=	Qa < ` G=
bool / _Bool	1	1
unsigned char / char	1	1
unsigned short / short	2	2
unsigned int / int	4	4
unsigned long / long	4	4
unsigned long long / long long	8	8
C+ ^ 3	4	4
float	4	4
double	8	8
long double	16	16

Qc ° » ¶ _ ABI ^ 3 Y char ¼b 6' ½¾ ^ 3 P

ELF u` ٪

< GÀÁ#c c ELF u` ٪Q1 / ´ | Od ÃHÃ d | • < Q SysV gABI P

EI_CLASS: ELF ٪ ´ |

EI_CLASS	Äe.	Åd
ELFCLASS32	1	32 Æ ELF ´
ELFCLASS64	2	64 Æ ELF ´

e_machi ne: ÇÈ f " ID

LoongArch (258)

e_fla gs: ABI ^ 3 É • < ` g

[31: 8] Æ	[7: 6] Æ	[5: 3] Æ	[2: 0] Æ
< 8 9 =	ABI • <	ABI h Ê Ë Ì	¶ _ ABI ^ 3

e_fla gs[7: 0] Í Î ` g Ĩ ELF u` ٪. / Q ABI ^ 3 P

Table 5. ¶ _ ABI ^ 3 ` g

¶ _ ABI 2 3	Äe. (e_fla gs[2: 0])	Åd
	0x0	8 9.
l p64s	0x1	. / 64 Æ 1 / +, - É, - HY ; ² µ³ 3 LP64 < l ong É C + ^ 3 i Ð 364 Æ Y i nt 332 Æ =
l p64f	0x2	. / 64 Æ 1 / +, - Y 32 Æ K L +, - É, - HY ; ² µ³ 3 LP64 < l ong É C + ^ 3 i Ð 364 Æ Y i nt 332 Æ =
l p64d	0x3	. / 64 Æ 1 / +, - Y 64 Æ K L +, - É, - HY ; ² µ³ 3 LP64 < l ong É C + ^ 3 i Ð 364 Æ Y i nt 332 Æ =
	0x4	8 9.
i l p32s	0x5	. / 32 Æ 1 / +, - É, - HY ; ² µ³ 3 ILP32 < i nt Y l ong É C + ^ 3 i Ð 332 Æ =
i l p32f	0x6	. / 32 Æ 1 / +, - Y 32 Æ K L +, - É, - HY ; ² µ³ 3 ILP32 < i nt Y l ong É C + ^ 3 i Ð 332 Æ =
i l p32d	0x7	. / 32 Æ 1 / +, - Y 64 Æ K L +, - É, - HY ; ² µ³ 3 ILP32 < i nt Y l ong É C + ^ 3 i Ð 332 Æ =

Table 6. ABI header

ABI header	Header (e_flags[5:3])	Ad
base	0x0	¼bYÑheader
	0x1 - 0x7	89.

e_flags[7:6] header ELF u` ¿ %o. / Q ABI • < P

Table 7. ABI • < header

ABI • <	Header	Öi
v0	0x0	" • ^ ' , Ó} NdQÖOÆ^3
v1	0x1	Ö89
	0x2 0x3	89.

ÔÖÆ ^ 3

Table 8. ELF ÔÖÆ ^ 3

Äe .	23	Öi	Nd
0	R_LARCH_NON E		
1	R_LARCH_32	j 2½¾@AoÖ	*(int32_t *) PC = RtAddr + A
2	R_LARCH_64	j 2½¾@AoÖ	*(int64_t *) PC = RtAddr + A
3	R_LARCH_REL ATIVE	µkj 2¤l @A×¥	*(void **) PC = B + A
4	R_LARCH_COP Y	Em- ØÛ; ² j 2ÚÛ	memcpy (PC, RtAddr, sizeof (sym))
5	R_LARCH_JUM P_SLOT	PLT Ûn " •	i ^ Ç=E Od
6	R_LARCH_TLS _DTPMOD32	TLS-GDj 2ÔÖÆ" •	*(int32_t *) PC = ID of module defini ng sym
7	R_LARCH_TLS _DTPMOD64	TLS-GDj 2ÔÖÆ" •	*(int64_t *) PC = ID of module defini ng sym
8	R_LARCH_TLS _DTPREL32	TLS-GDj 2ÔÖÆ" •	*(int32_t *) PC = DTV- relative offset for sym
9	R_LARCH_TLS _DTPREL64	TLS-GDj 2ÔÖÆ" •	*(int64_t *) PC = DTV- relative offset for sym
10	R_LARCH_TLS _TPREL32	TLS-IEj 2ÔÖÆ" •	*(int32_t *) PC = T
11	R_LARCH_TLS _TPREL64	TLS-IEj 2ÔÖÆ" •	*(int64_t *) PC = T
12	R_LARCH_I RE LATIVE	< @oÝÛnoÖ	*(void **) PC = (((void)*)()) (B + A)) ()
Éj 2RÝ - 89D			
20	R_LARCH_MAR K_LA	` g la.abs ÞCß	J 2ÚÛ½¾pQ@A
21	R_LARCH_MAR K_PCREL	` g àž` qÛn	J 2ÚÛ½¾@A á f.
22	R_LARCH_SOP _PUSH_PCREL	ˆ ½¾Ž Q@Ar ,	push (S - PC + A)
23	R_LARCH_SOP _PUSH_ABSOL UTE	ˆ : ; â pQ@Ar ,	push (S + A)
24	R_LARCH_SOP _PUSH_DUP	sã , t äå	opr1 = pop (), push (opr1), push (opr1)

Äe .	23	Ò _i	Nd
25	R_LARCH_SOP_PUSH_GPREL	$\frac{1}{2}\frac{3}{4}Q\text{ GOT } \text{æD} \text{á } f \cdot r ,$	push (G)
26	R_LARCH_SOP_PUSH_TLS_TPREL	$\frac{1}{2}\text{ TLS-LE } \text{á } f \cdot r ,$	push (T)
27	R_LARCH_SOP_PUSH_TLS_GOT	$\frac{1}{2}\text{ TLS-IE } \text{á } f \cdot r ,$	push (IE)
28	R_LARCH_SOP_PUSH_TLS_GD	$\frac{1}{2}\text{ TLS-GD } \text{á } f \cdot r ,$	push (GD)
29	R_LARCH_SOP_PUSH_PLT_PCREL	$\frac{1}{2}\frac{3}{4}\text{ PLT stub } Q@A \text{á } f \cdot r ,$	push (PLT - PC)
30	R_LARCH_SOP_ASSERT	$\text{ç} \bullet , \text{ t } \text{ä} \text{å} \text{3è}$	assert (pop ())
31	R_LARCH_SOP_NOT	$, \text{ t } \text{u} \text{é}$	push (!pop ())
32	R_LARCH_SOP_SUB	$, \text{ t } \text{u} \text{é}$	opr2 = pop (), opr1 = pop (), push (opr1 - opr2)
33	R_LARCH_SOP_SL	$, \text{ t } \text{u} \text{é}$	opr2 = pop (), opr1 = pop (), push (opr1 << opr2)
34	R_LARCH_SOP_SR	$, \text{ t } \text{u} \text{é}$	opr2 = pop (), opr1 = pop (), push (opr1 >> opr2)
35	R_LARCH_SOP_ADD	$, \text{ t } \text{u} \text{é}$	opr2 = pop (), opr1 = pop (), push (opr1 + opr2)
36	R_LARCH_SOP_AND	$, \text{ t } \text{u} \text{é}$	opr2 = pop (), opr1 = pop (), push (opr1 & opr2)
37	R_LARCH_SOP_IF_ELSE	$, \text{ t } \text{u} \text{é}$	opr3 = pop (), opr2 = pop (), opr1 = pop (), push (opr1 ? opr2 : opr3)
38	R_LARCH_SOP_POP_32_S_10_5	$C\beta g \text{ê} ; \text{ÔO}\text{Æ}$	opr1 = pop (), (*(ui nt32_t *) PC) [14 ... 10] = opr1 [4 ... 0] $\vee 5 \text{Æ}' \frac{1}{2}\frac{3}{4}; \text{ë} \text{ì } w x \text{í } Z$
39	R_LARCH_SOP_POP_32_U_10_12	$C\beta g \text{ê} ; \text{ÔO}\text{Æ}$	opr1 = pop (), (*(ui nt32_t *) PC) [21 ... 10] = opr1 [11 ... 0] $\vee 12 \text{Æ}\tilde{N} \frac{1}{2}\frac{3}{4}; \text{ë} \text{ì } w x \text{í } Z$

Äe.	23	Òi	Nd
40	R_LARCH_SOP _POP_32_S_1 0_12	Cßgê; ÔOÆ	opr1 = pop (), (*(ui nt32_t *) PC) [21 ... 10] = opr1 [11 ... 0] v 12Æ' ½¾; ëì wx í Z
41	R_LARCH_SOP _POP_32_S_1 0_16	Cßgê; ÔOÆ	opr1 = pop (), (*(ui nt32_t *) PC) [25 ... 10] = opr1 [15 ... 0] v 16Æ' ½¾; ëì wx í Z
42	R_LARCH_SOP _POP_32_S_1 0_16_S2	Cßgê; ÔOÆ	opr1 = pop (), (*(ui nt32_t *) PC) [25 ... 10] = opr1 [17 ... 2] v 18Æ' ½¾; ëì É4` GQa wx í Z
43	R_LARCH_SOP _POP_32_S_5 _20	Cßgê; ÔOÆ	opr1 = pop (), (*(ui nt32_t *) PC) [24 ... 5] = opr1 [19 ... 0] v 20Æ' ½¾; ëì wx í Z
44	R_LARCH_SOP _POP_32_S_0 _5_10_16_S2	Cßgê; ÔOÆ	opr1 = pop (), (*(ui nt32_t *) PC) [4 ... 0] = opr1 [22 ... 18], (*(ui nt32_t *) PC) [25 ... 10] = opr1 [17 ... 2] v 23Æ' ½¾; ëì É4` GQa wx í Z
45	R_LARCH_SOP _POP_32_S_0 _10_10_16_S 2	Cßgê; ÔOÆ	opr1 = pop (), (*(ui nt32_t *) PC) [9 ... 0] = opr1 [27 ... 18], (*(ui nt32_t *) PC) [25 ... 10] = opr1 [17 ... 2] v 28Æ' ½¾; ëì É4` GQa wx í Z
46	R_LARCH_SOP _POP_32_U	Cß × ¥	(*(ui nt32_t *) PC) = pop () v 32Æ' ½¾; ëì wx í Z
47	R_LARCH_ADD 8	8Æî @¤S	*(int8_t *) PC += S + A
48	R_LARCH_ADD 16	16Æî @¤S	*(int16_t *) PC += S + A

Äe.	23	Ò _i	Nd
49	R_LARCH_ADD 24	24 Æî @x S	*(int24_t *) PC += S + A
50	R_LARCH_ADD 32	32 Æî @x S	*(int32_t *) PC += S + A
51	R_LARCH_ADD 64	64 Æî @x S	*(int64_t *) PC += S + A
52	R_LARCH_SUB 8	8 Æî @y S	*(int8_t *) PC -= S + A
53	R_LARCH_SUB 16	16 Æî @y S	*(int16_t *) PC -= S + A
54	R_LARCH_SUB 24	24 Æî @y S	*(int24_t *) PC -= S + A
55	R_LARCH_SUB 32	32 Æî @y S	*(int32_t *) PC -= S + A
56	R_LARCH_SUB 64	64 Æî @y S	*(int64_t *) PC -= S + A
57	R_LARCH_GNU _VTI NHERI T	GNU C++ vtable " •	
58	R_LARCH_GNU _VTENTRY	GNU C++ vtable " •	

j 2RÝ - i ð

Table 9. `ñj 2RÝ - i ððæ_

ABI ^ 3	ABI hÊĚĬ	Ó} È z / C {	Glibc j 2RÝ - i ð
l p64d	base	Linux, Glibc	/l i b64/l d-l i nux-l oongarch-l p64d. so. 1
l p64f	base	Linux, Glibc	/l i b64/l d-l i nux-l oongarch-l p64f. so. 1
l p64s	base	Linux, Glibc	/l i b64/l d-l i nux-l oongarch-l p64s. so. 1
i l p32d	base	Linux, Glibc	/l i b32/l d-l i nux-l oongarch-i l p32d. so. 1
i l p32f	base	Linux, Glibc	/l i b32/l d-l i nux-l oongarch-i l p32f. so. 1
i l p32s	base	Linux, Glibc	/l i b32/l d-l i nux-l oongarch-i l p32s. so. 1