```
walterzh@walterzh-ThinkPad-T440p:~/gerrit/linux-module/kernel/drivers/i2c/busses$ modinfo i2c-pxa.ko

filename:       /home/walterzh/gerrit/linux-module/kernel/drivers/i2c/busses/i2c-pxa.ko

alias:          platform:pxa2xx-i2c

license:        GPL

alias:          platform:ce4100-i2c

alias:          platform:pxa3xx-pwri2c

alias:          platform:pxa2xx-i2c

alias:          of:N*T*Cmrvl,mmp-twsi*

alias:          of:N*T*Cmrvl,pwri2c*

alias:          of:N*T*Cmrvl,pxa-i2c*

depends:

intree:         Y

vermagic:       3.18.7-yocto-standard SMP preempt mod_unload ARMv7 p2v8
```

modinfo就是show在.modinfo section中的内容。

```
$ arm-linux-gnueabi-objdump -s i2c-pxa.ko


Contents of section .modinfo:

 0000 616c6961 733d706c 6174666f 726d3a70  alias=platform:p

 0010 78613278 782d6932 63006c69 63656e73  xa2xx-i2c.licens

 0020 653d4750 4c000000 616c6961 733d706c  e=GPL...alias=pl

 0030 6174666f 726d3a63 65343130 302d6932  atform:ce4100-i2

 0040 6300616c 6961733d 706c6174 666f726d  c.alias=platform
```

```
0050 3a707861 3378782d 70777269 32630061  :pxa3xx-pwri2c.a
0060 6c696173 3d706c61 74666f72 6d3a7078  lias=platform:px
0070 61327878 2d693263 00616c69 61733d6f  a2xx-i2c.alias=o
0080 663a4e2a 542a436d 72766c2c 6d6d702d  f:N*T*Cmrvl,mmp-
0090 74777369 2a00616c 6961733d 6f663a4e  twsi*.alias=of:N
00a0 2a542a43 6d72766c 2c707772 6932632a  *T*Cmrvl,pwri2c*
00b0 00616c69 61733d6f 663a4e2a 542a436d  .alias=of:N*T*Cm
00c0 72766c2c 7078612d 6932632a 00000000  rvl,pxa-i2c*....
00d0 64657065 6e64733d 00696e74 7265653d  depends=.intree=
00e0 59007665 726d6167 69633d33 2e31382e  Y.vermagic=3.18.
00f0 372d796f 63746f2d 7374616e 64617264  7-yocto-standard
0100 20534d50 20707265 656d7074 206d6f64   SMP preempt mod
0110 5f756e6c 6f616420 41524d76 37207032  _unload ARMv7 p2
0120 76382000                             v8 .
```

i2c-pxa.ko module的alias

1. MODULE_ALIAS("platform:pxa2xx-i2c");

in include/linux/mdoule.h

```
#define MODULE_ALIAS(_alias) MODULE_INFO(alias, _alias)
```

```
/* Generic info of form tag = "info" */
```

```
#define MODULE_INFO(tag, info) __MODULE_INFO(tag, tag, info)


#define __MODULE_INFO(tag, name, info)                                \
static const char __UNIQUE_ID(name)[]                                 \
  __used __attribute__((section(".modinfo"), unused, aligned(1)))  \
  = __stringify(tag) "=" info
```

2. i2c-pxa.ko的其他alias

```
alias:          platform:ce4100-i2c
alias:          platform:pxa3xx-pwri2c
alias:          platform:pxa2xx-i2c
```

==>

```
static const struct platform_device_id i2c_pxa_id_table[] = {
        { "pxa2xx-i2c",         REGS_PXA2XX },
        { "pxa3xx-pwri2c",   REGS_PXA3XX },
        { "ce4100-i2c",         REGS_CE4100 },
        { },
};
MODULE_DEVICE_TABLE(platform, i2c_pxa_id_table);
```

```
alias:          of:N*T*Cmrvl,mmp-twsi*
```

```
alias:          of:N*T*Cmrvl,pwri2c*
alias:          of:N*T*Cmrvl,pxa-i2c*
```

==>

```
static const struct of_device_id i2c_pxa_dt_ids[] = {
	{ .compatible = "mrvl,pxa-i2c", .data = (void *)REGS_PXA2XX },
	{ .compatible = "mrvl,pwri2c", .data = (void *)REGS_PXA3XX },
	{ .compatible = "mrvl,mmp-twsi", .data = (void *)REGS_PXA2XX },
	{}
};
MODULE_DEVICE_TABLE(of, i2c_pxa_dt_ids);
```

```
/* Creates an alias so file2alias.c can find device table. */
#define MODULE_DEVICE_TABLE(type, name)					\
  extern const struct type##_device_id __mod_##type##__##name##_device_table \
  __attribute__ ((unused, alias(__stringify(name))))
```

==>

```
MODULE_DEVICE_TABLE(of, i2c_pxa_dt_ids);

  extern const struct of_device_id __mod_of__i2c_pxa_dt_ids_device_table \
  __attribute__ ((unused, alias(__stringify(i2c_pxa_dt_ids))))
```

???

从comments看，是为scripts/mod/file2alias.c准备的。

从i2c_pxa_dt_ids[]中的"mrvl,mmp-twsi"怎么得到of:N*T*Cmrvl,mmp-twsi*的？

file2alias.c做的活吗？