



为 uCLinux 交叉编译 SDL

(陈云川 ycb2084@163.com UESTC,CD 2007 年 8 月 16 日)

1 prepare

首先，要安装以 arm-elf-gcc 为代表的 uCLinux 工具链，我采用的 arm-elf-gcc 版本是 2.95.3；其次要准备好 SDL 源代码包，我采用的 SDL 源代码版本是 1.2.5；最后，你得自己在开发板上把 uCLinux 跑起来，顺便说一下，我所用的开发板是 ARM7。

假定解压后 SDL 源代码的根目录为/home/rockins/SDL/SDL-1.2.5/，安装目录设置为/home/rockins/SDL/SDL-1.2.5/uclinux-build。在你的系统上，这两个目录应该根据具体情况而有所不同。

2 configure

由于我用的开发板是 ARM7，因此很多功能都用不了，这些鸡肋功能都在 configure 的阶段直接被禁用了——至于哪些功能可能用不了则是我大胆猜测的。鉴于此，这些被禁用的功能中可能有些是被“误杀”的，但是无论如何，我秉持的观点是“宁可错杀一千，不可放过一个”。最终的结果是仅剩下一些最基本的功能被保留了下来。下面的命令是在控制台上运行 configure 脚本：

```
./configure --prefix=/home/rockins/SDL/SDL-1.2.5/uclinux-build --enable-shared=no --enable-audio=no --enable-joystick=yes --enable-cdrom=no --enable-threads=no --enable-timers=yes --enable-endian=yes --enable-file=yes --enable-esd=no --enable-arts=no --enable-nas=no --enable-diskaudio=no --enable-mintaudio=no --enable-nasm=no --enable-video-x11=no --enable-dga=no --enable-video-x11-vm=no --enable-video-x11-dgmouse=no --enable-video-x11-xv=no --enable-video-x11-xinerama=no --enable-video-x11-xme=no --enable-video-dga=no --enable-video-photon=no --enable-video-directfb=no --enable-video-ps2gs=no --enable-video-xbios=no --enable-video-gem=no --enable-video-opengl=no --enable-pth=no --enable-pthreads=no --enable-pthread-sem=no
```

简要说明：

- --prefix=/home/rockins/SDL/SDL-1.2.5/uclinux-build

指定编译后的安装目录，根据需要进行设置。

- --enable-shared=no

目前还无法编译成动态共享库，故需要禁用动态共享库。



3 make

编译 SDL 只需在控制台上输入如下命令即可：

```
make -e CC=arm-elf-gcc CXX=arm-elf-g++ CCAS=arm-elf-gcc CCLD=arm-elf-gcc RANLIB=arm-elf-ranlib OBJDUMP=arm-elf-objdump AR=arm-elf-ar
```

这些参数实际上控制着 Makefile 中的相应变量。用 arm-elf 工具链替换之后，就是用 arm-elf 的工具对 SDL 源代码进行交叉编译。生成的 SDL 函数库（在本文中是静态链接库）就可以和程序代码链接，并在 uCLinux 平台上运行。

4 test

在/home/rockins/SDL/SDL-1.2.5/test/目录下是用来测试 SDL 的一些测试例程。本文选择的测试例程是 testver.c，其源代码如下所示：

```
/* Test program to compare the compile-time version of SDL with the linked
   version of SDL
*/

#include <stdio.h>

#include "SDL.h"
#include "SDL_byteorder.h"

int main(int argc, char *argv[])
{
    SDL_version compiled;

    /* Initialize SDL */
    if ( SDL_Init(0) < 0 ) {
        fprintf(stderr, "Couldn't initialize SDL: %s\n",SDL_GetError());
        exit(1);
    }

#ifdef DEBUG
    fprintf(stderr, "SDL initialized\n");
#endif

    #if SDL_VERSION_ATLEAST(1, 2, 0)
        printf("Compiled with SDL 1.2 or newer\n");
    #else
        printf("Compiled with SDL older than 1.2\n");
    #endif
}
```



```
SDL_VERSION(&compiled);
printf("Compiled version: %d.%d.%d\n",
      compiled.major, compiled.minor, compiled.patch);
printf("Linked version: %d.%d.%d\n",
      SDL_Linked_Version()->major,
      SDL_Linked_Version()->minor,
      SDL_Linked_Version()->patch);
printf("This is a %s endian machine.\n",
      (SDL_BYTEORDER == SDL_LIL_ENDIAN) ? "little" : "big");
SDL_Quit();
return(0);
}
```

采用下列命令编译此程序：

```
arm-elf-gcc -o testver testver.c -I/home/rockins/SDL/SDL-1.2.5/uclinux-build/include/SDL/ -L
/home/rockins/SDL/SDL-1.2.5/uclinux-build/lib/ -elf2flt -lSDL
```

特别需要注意的是：在编译此程序时一定要加上-elf2flt 选项，否则会出现如下错误：

```
undefined reference to `__CTOR_LIST__'
undefined reference to `__DTOR_LIST__'
```

实际上，不只是编译testver.c程序时要这样，编译其它运行在uCLinux平台上的程序时，都要加上这个选项，否则就会出现上述错误。原因简述如下：通常，在libgcc.a中是有__CTOR_LIST__和__DTOR_LIST__这两个符号的。然而，在uCLinux的toolchain中，由于某种目的，这两个符号被删掉了，因此直接编译时就会出现找不到符号引用的错误。所幸的是，在elf2flt的link script中提供了这样两个符号。因此，在采用uCLinux工具链编译可执行程序的时候，一定要加上-elf2flt选项【以上说明来自www.chinaunix.net某个帖子，但是具体地址已经找不到了】。

言归正传，在编译好了 testver 程序之后，通过 ftp 或者 nfs 或者诸如此类的手段将其下载到开发板上，执行之，你将看到程序打印出 SDL 编译时和执行时采用的版本信息。这表明你的 SDL 静态链接库已经编译成功了。