BUILDING notes for code in *The Linux Programming Interface*

```
PLEASE TAKE NOTE!
=================

If you have difficulty building the programs, then the problem is most
likely one of the following:

a) A configuration issue on your system (e.g., do you have the 'libacl'
   library installed?).
b) You are using a system with an older Linux kernel or an older
   version of glibc. If this is so, your system may not provide a
   more recent system call or library function, or the glibc headers
   may not provide a needed function declaration or constant
   definition.

I can't really help with fixing these problems. However, do take
a look at the online code FAQ (http://man7.org/tlpi/code/faq.html)
for some frequently asked questions about the code. You may find
the answer to your problem there.

If, after ensuring that the problem is not a configuration issue on your
system (did you try compiling the program(s) on a different system?) and
checking the FAQ, you still have a problem *and* you have a solution, then
please do let me know, and I will publish the fix on the web site.

Unpacking the code tarball
==========================

Unpacking the tarball with the command

        tar xfz tlpi-YYMMDD-xxxx.tar.gz

will create a directory tree 'tlpi-xxxx' containing all of the source code
("xxxx" will be either "dist" or "book" depending on which version of the
source code tarball you downloaded).


Building the programs on Linux
==============================

(For instructions on building the programs on other UNIX
implementations, see the notes lower down in this file.)

The methods described below involve the use of 'make'. If you need
more information on 'make', try the command 'man make' or 'info make'.


Method A - Building all programs in all directories
---------------------------------------------------

Go into the 'tlpi' subdirectory, and type 'make':

        cd tlpi-xxxx
        make

This will build all programs in all subdirectories.


Method B - Build programs in individual directories
---------------------------------------------------

1) First, build the library used by all programs:

        cd lib
        make            # This will make libtlpi.a and place
                        # it in the parent directory

2) Build programs using the 'Makefile' in each subdirectory.

In each subdirectory, there is a file called 'Makefile' that can be
used with the 'make' command to build the programs in that directory.
To build a particular program you can simply say:

        make progname

where 'progname' is one of the C (main) programs in the directory.
Thus, to build the executable for the program 'copy.c' you would use
the command:

        make copy

Each makefile also contains the following special targets (which are
named according to the usual 'make' conventions):

all     This target will build all programs in the subdirectory.
        Simply enter the command:
```

```
                make all

clean   This target cleans up, by removing all executables and object
        files created by the makefile:

                make clean


Building the programs on other UNIX implementations
===================================================

I have gone to some effort to ensure that, where possible, the programs
work on other UNIX implementations also.

Obviously, some of the example programs employ Linux-specific features,
and won't compile on other UNIX implementations. As a first indication
of what works on Linux, and what may work elsewhere, look in the
makefiles in each directory. Most of the makefiles define two macros:

LINUX_EXE       These are programs that probably won't work
                on anything other than Linux.

GEN_EXE         These programs may compile on other UNIX implementations.
                I say "may", because of course not all implementations
                provide exactly the same features. The presence of a
                program in this list indicates that it compiles and runs
                on at least some UNIX implementations other than Linux.


Instructions
------------

1. Edit Makefile.inc in the 'tlpi' root directory to modify the
   definitions of the CFLAGS and LDLIBS macros (and possibly other
   macros depending on your version of make(1)) as appropriate.
   Probably you'll need to define CFLAGS as follows:

        CFLAGS += -g -I${TLPI_INCL_DIR}

   The setting of LDLIBS is a bit harder to determine. As well as
   listing the libary for this book, you should include '-l' linker
   options for any other libraries that the programs may need.
   For example, the following is suitable on Solaris 8:

        LDLIBS = ${TLPI_LIB} -lsocket -lnsl -lrt -ldl -lm -lresolv

   **** NOTE:   Under the 'tlpi' root directory you'll find a few sample
                replacement files for 'Makefile.inc', named
                'Makefile.inc.*'. You may be able to simply copy the
                appropriate file to 'Makefile.inc'.

2. Try the following to build all of the GEN_EXE programs:

        cd tlpi-xxxx
        make -k allgen

   The '-k' option tells 'make' to build as much as possible, so that if
   a particular program won't compile, 'make' goes on to attempt to
   build the remaining programs.
```