APPENDIX G

Instructions for Recompiling the PDP Programs

Complete source code for all the programs described in this book is provided in the archive *src.arc*. If you have Version 3.0 (or higher) of the Microsoft C compiler, you should be able to modify the programs and recompile them on your PC. If you have a UNIX system (Berkeley 4.2 or higher) with the standard UNIX C compiler, you should also be able to recompile the programs to run on that system. We first provide a general inventory of the components of the PDP software and of the dependencies among these components. Then we describe the procedure for recompiling for the PC with Microsoft C. Following this we briefly describe how to set up the PDP software on UNIX systems. You are free to try to use other C compilers, but with others you are completely on your own. There is every reason to expect that some tinkering will be required to recompile the software with non-UNIX C compilers other than Microsoft C.

Components of the PDP Software

The software includes seven executable programs: **aa, bp, cl, cs, ia, iac,** and **pa**. For each of these programs, there is a source file with the same name (e.g., *aa.c*). In addition, there are certain other source files that several programs share. The object files that all programs share are grouped into one library file called *libpc.a*. This library file is made up of the compiled versions of the routines from the files *command.c, display.c, general.c, io.c, main.c, patterns.c, template.c,* and *variable.c*. The software also includes the two utility programs: **plot** and **colex**; each of these is constructed from a single corresponding *.c* file.

If you change any source file, you should recompile and relink all object and executable files that depend upon that source file. The following dependency list shows which executables depend upon which source files. Where *libpc.a* is shown, all eight source files in the library are included in the dependency.

aa:	*aa.c, libpc.a*
bp:	*bp.c, weights.c, libpc.a*
cl:	*cl.c, libpc.a*
cs:	*cs.c, weights.c, libpc.a*
ia:	*ia.c, iaaux.c, iatop.c, libpc.a*
iac:	*iac.c, weights.c*
pa:	*pa.c, weights.c, libpc.a*
plot:	*plot.c*
colex:	*colex.c*

Note also that there are many header files (with names ending in *.h*) in the PDP software package. These files often contain declarations that are used in several different modules. This is particularly true for the header files associated with the modules in *libpc.a*. If one of these files is modified, it is prudent to recompile all modules that include this file. The following list indicates which *.h* files are included in each *.c* file:

general.h:	*display.h*
aa.c:	*general.h, aa.h, variable.h, patterns.h, command.h*
bp.c:	*general.h, bp.h, variable.h, weights.h, patterns.h, command.h*
cl.c:	*general.h, variable.h, patterns.h, command.h, cl.h*
command.c:	*general.h, io.h, command.h*
cs.c:	*general.h, cs.h, variable.h, command.h, patterns.h, weights.h*
display.c:	*general.h, io.h, variable.h, template.h, weights.h, command.h*
general.c:	*general.h, command.h, variable.h*
ia.c:	*ia.h, io.h, general.h*
iaaux.c:	*ia.h*
iac.c:	*general.h, iac.h, variable.h, command.h, weights.h, patterns.h*
iatop.c:	*general.h, cs.h, variable.h, command.h, ia.h*
io.c:	*io.h*
main.c:	*general.h, variable.h, command.h, patterns.h*
pa.c:	*general.h, pa.h, variable.h, weights.h, patterns.h, command.h*
patterns.c:	*general.h, command.h, variable.h, patterns.h*
template.c:	*general.h, command.h, variable.h, display.h, template.h*

variable.c:	*general.h, variable.h, command.h, patterns.h, weights.h*
weights.c:	*general.h, command.h, weights.h, variable.h*

Note that all files that include *general.h* also implicitly include *display.h*.

To Recompile for a PC Using Microsoft C

Three batch files are provided in the *src* directory to aid in compiling and linking: *compile.bat, makelib.bat,* and *pdplink.bat*. These files can be executed as though they were programs.

The *compile* batch file includes the command to compile a source file (*.c*) into an object file (*.obj*). The first step in creating an executable (*.exe*) file is to recompile all of the source files upon which it depends, including the files in *libpc.a*. To recompile all the source files, execute the following command in the directory that contains the files:

```
compile all
```

This will produce a *.obj* file for each *.c* file in that directory. To compile only one source file, use its name instead of *all* when giving the *compile* command. Thus, to compile *aa.c* you would enter

```
compile aa.c
```

This will produce a file called *aa.obj* in that directory. If there are errors in your source code, the compiler may abort the command file and display the error messages.

Once all of the necessary object files are created, the *libpc.a* file can be built. The command file for building the library is executed as follows:

```
makelib
```

This will create a file called *libpc.a* in the current directory from the eight object files, which should be in the same directory. Whenever you recompile any of the eight programs in *libpc.a*, you should use the *makelib* command again to update the library. (The *makelib* command requires all the object files to be present, so it is best to keep these files around while you are actively involved in modifying the programs. Once you stop making changes, you can delete the *.obj* files to save space.)

The final stage in compiling is linking. The command file *pdplink.bat* will link the necessary files to create each of the executables. To link the object files for a particular program, enter *pdplink* with the program name as argument. Thus,

```
pdplink bp
```

will link the files *bp.obj* and *weights.obj* with the library *libpc.a* to create the executable *bp.exe*. The *pdplink* command will also work with **plot** or **colex**. To create executables for all seven PDP simulation programs, use the following command:

```
pdplink all
```

Note that *pdplink all* does not link **colex** and **plot**; these must be linked individually.

If you wish to recompile and relink all of the programs at once, use the following three commands:

```
compile all
makelib
pdplink all
```

Once you have created new executables, you will want to move these files into the appropriate working directories, to be used with the relevant *.tem* and *.str* files. The MS-DOS copy utility can be used to do this.

Instructions for Setting Up the PDP Software on UNIX Systems

For UNIX systems, we suggest that you set up a parent directory system for the PDP software and copy the extracted contents of each of the *.arc* files into a separate subdirectory of the parent, giving the subdirectory the same name as the archive. For example, if your parent directory were called */usr/yourname/pdp*, you would put the contents of *aa.arc* into a subdirectory of this directory called */usr/yourname/pdp/aa*. The only files that you will not want to include in this directory system are the *.exe* files, since these will only run on PCs. You would also create a subdirectory called */usr/yourname/pdp/src* containing the source files and other materials necessary to recompile the package from the *src.arc* file.

Once the directories have been set up, you will want to change directories to the *src* subdirectory. It is an easy matter to recompile all the programs because we have supplied a *makefile*. This file is used by the UNIX **make** program to manage the PDP software. To compile all of the PDP simulation programs, you need only execute the following command:

```
make
```

To compile a single program, simply give *make* the name of that program as an argument. For example, to recompile the **aa** program, enter

```
make aa
```

This form also works with the **plot** and **colex** programs; they are not updated if **make** is executed with no arguments.

In either case, **make** will check the *makefile* to see which source files need to be recompiled and will recompile them. It will update *libpc.a* if necessary. And it will link the necessary object modules together to create the necessary executable files. The supplied makefile places the seven PDP executables in directories that are on the same level as the source directory and have the same name as the executable. For example, if the *src* directory is */usr/yourname/pdp/src* then the **aa** executable would be placed in the directory */usr/yourname/pdp/aa*. If you have set up subdirectories for each program as suggested above, this will all work fine. If you have chosen to organize the directories differently, the *makefile* can be modified to change where each program is placed. For each program there is a variable that specifies the destination directory for the executable version of the program. The names of these variables are uppercase and consist of the program name followed by *DEST*. Thus for **aa**, there is a line in the *makefile* that looks like this:

```
AADEST = ../aa/
```

The path name to the right of the equal sign can be replaced by any other valid UNIX path name. Once it is, **aa** will be stored in the directory specified by the path. Thus

```
AADEST = /usr/foo/pc/bin/
```

would cause **make** to put the **aa** executable in the directory */usr/foo/pc/bin*.