

```
cat part* > binutils.tar.gz
gunzip binutils.tar.gz
tar xf binutils.tar
rm part* binutils.tar
```

Assemble the parts into a single compressed tar file, uncompress it, and untar it. If you're short on space, you can get rid of some extra files now.

```
cd binutils-2.8.1
configure
```

Configure the software for building on your system.

```
cd bfd
make
cd ../libiberty
make
```

Build the software. In the bfd directory you should have libbfd.a, and in the libiberty directory you should have libiberty.a

7.0 Building the Simulator, Elsie, and the Benchmarks

The distribution directory contains a **README** file that explains how to reconstruct the WWT-II source tree from the distributed files. Following that procedure will result in a subdirectory called *wwt_ii*.

We have attempted to write the *Makefiles* which build the simulator in a way which avoids dependences on our local environment. To get this to work, you will need to set some environment variables and make sure certain programs are in your search path before building WWT-II. We recommend doing this in your login script so that things will work properly in case you need to start the build more than once.

You need to tell WWT-II's *Makefiles* where your *wwt_ii* directory is by setting the environment variable **WWT_ROOT**.

Section 6.0 explains how to obtain and build the GNU Binary File Utilities in case you don't already have them installed. Regardless of whether you had them previously or built them specially for use by WWT-II, you need to tell WWT-II's *Makefiles* where they are by setting the environment variable **BINUTILS**. This should be a directory name which ends with something like `"/binutils-2.8.1"`.

WWT-II's target programs are written using the **PARMACS** parallel programming directives. The target program source files contain **PARMACS** macros which must be processed using the **GNU m4** macro processor, (other versions of *m4* will not work properly). You must tell WWT-II's *Makefiles* where to locate the *m4* macro processor by setting the environment variable **GNU_M4**.

Some parts of the build procedure are accomplished with **PERL** scripts. The *Makefiles* invoke **PERL** directly, so you need to make sure that **PERL** is in your search path before beginning the build procedure.

Once you have your search path and environment variables set up properly, you can just go into your *wwt_ii* directory and type ***gmake -f Makefile.wwt2***. By *gmake* we mean whatever command you use to invoke the **GNU** version of *make* in your environment, (other versions of *make* will not work).

While the simulator is being built, you will see a few warnings generated by the *Make* and the compilers. So that you may determine whether the warnings you see are normal, we have included the output of a complete build in this distribution. This file is located at *wwt_ii/docs/wwt2/make.out*.

8.0 Getting Help

To request help with installation of WWT-II, send email to wwt@cs.wisc.edu.

- The only networking software supported at this time in LAM (Lanai Active Messages). This software is available from the Berkeley NOW project. They are on the world wide web at <http://now.cs.berkeley.edu/>.

5.0 Detailed Description of Build Environment

Our build environment makes heavy use of the GNU software building tools available from the Free Software Foundation. This distribution is set up to use

- GNU *Make*
- GNU Compilers — *gcc* and *g++*
- GNU *Binary File Utilities*
- GNU Macro processor — *m4*

Explaining how to build the GNU *make* and compiler programs is beyond the scope of this document. If you don't already have those on your system, you should contact the *Free Software Foundation* for help. They can be reached via the world wide web at <http://www.gnu.ai.mit.edu/home.html>.

Don't despair if you don't already have the GNU *Binary File Utilities*, they're easy to get and build. We'll tell you how in the next section.

6.0 Obtaining and Building the GNU *Binary File Utilities*

Due to some of the restrictions placed on GNU software by their license (copy-left), we cannot include the *Binary File Utilities* in the WWT-II distribution. These utilities are available from the Free Software Foundation's GNU Project, via ftp. The files are available at any of several "mirror sites" located around the world. A current list of mirror sites is available at <http://www.gnu.ai.mit.edu/server/list-mirrors.html>.

We recently obtained and built a copy to get an idea what is involved. We got our copy from *ftp.uu.net* and built it using the following procedure:

mkdir binutils cd binutils	Create a directory to hold the utilities and change directory into it
ftp ftp.uu.net anonymous <your email address>	Give "anonymous" as your user name, and give your email address as your password.
cd archive/systems/gnu/binutils-2.8.1.tar.gz-split	Change to the remote directory which contains the binutils software
prompt binary hash	Set up your ftp client to a. not prompt you between fetching multiple files, b. use binary transfer mode, and c. show progress by printing a hash mark after transferring every file block.
mget * quit	The remote directory contains a compressed tar file, which has been split up into several smaller files for convenience in transferring over the net. This command says to fetch them all.
sum part* > sums diff README sums	Check to see that all the files got transferred correctly. The proper checksums for all the pieces of the compressed tar file are in README.

1.0 Introduction

The Wisconsin Wind Tunnel - 2 (WWT-II) is a parallel execution driven discrete event simulator for parallel machine architectures. It runs on a variety of SPARC¹ based Sun² systems, and simulates a wide range of parallel systems based on SPARC-like CPUs

This document explains the hardware and software requirements for building and running the simulator, and tells how to proceed in building it and its related software. A separate document called the **Wisconsin Wind Tunnel II: User's Guide** tells how to use the simulator once you have it built.

2.0 Hardware and Software Requirements

The simulator can run on a variety of SUN SPARC based machines, including single workstations, networks of workstations, SUN SMPS, and networks of SMPs. We have built this software in a SUN Solaris³ environment, mostly using GNU tools. We strongly recommend that you use the same GNU tools if you want to build this software.

3.0 Description of Distribution Tree

The simulator is distributed as a compressed tar file which is then split into 256KByte chunks. After you have unpacked the file, you will have a file system tree rooted at a directory called *wwt_ii*. The various subtrees are:

- **docs** — WWT-II related documents, including this one.
- **tzero** — Stuff related directly to the simulator.
 - **mbussim** — The actual simulator code.
 - **elsie** — A tool for preparing target binaries to run on the simulator.
- **EEL** — Executable Editing Library, needed by Elsie.
- **sam** — Synchronized Active Messages, a portable synchronization and messaging layer.
- **tempest** — Libraries and include files needed by target programs.
- **example** — A purely pedagogical target program which should be easy to understand and build.
- **benchmarks.src**⁴ — A few sample benchmark programs to get you started.

4.0 Description of Supported Host Environments

The simulator requires one of a very specific set of host environments. The supported host environments are:

- Single SUN Workstations
- SUN SMPs
- Networks of SUN Workstations (including SMPs)
 - The only network supported at this time is the *myrinet*. Information is available via the world wide web at <http://www.myri.com>.

1. SPARC is a registered trademark of SPARC International, Inc.
2. SUN is a registered trademark of SUN Microsystems.
3. SOLARIS is a registered trademark of SUN Microsystems.
4. During the build process, this directory gets moved to “benchmarks”.