

Southampton's Portable Occam Compiler (SPOC)

Version 1.1

Installation Notes

P5404 GP MIMD

Our Ref: GPMIMD:012/O2C/INSTALL

Document Revision 0.2

Mark Debbage (md@uk.ac.srf.pact) Mark Hill (mbh@uk.ac.srf.pact)
Sean Wykes (smw@uk.ac.soton.ecs) Denis Nicole (dan@uk.ac.soton.ecs)

March 1, 1994

Copyright 1994 University of Southampton

This software and the accompanying documentation are licenced free of charge with absolutely no warranty. Copyright of the software and documentation is retained by the University of Southampton.

Further copies of this software system may be made and distributed at will, provided that it is distributed in full with no modifications and that no charge is made other than media costs. Parties wishing to distribute modified versions of the package, including commercial products utilising components of the package, should contact the University for further clarification. Any derivative work must retain an acknowledgement to the University of Southampton and ESPRIT.

Since this software is licenced free of charge it is supplied without warranty of any kind, either expressed or implied. The entire risk as to the quality and performance of this software is with the user.

1 Introduction

This document accompanies SPOC (Southampton's Portable **occam** Compiler). It describes the system's installation and its use to generate, compile and execute a simple **occam** program.

Although the final intention is to provide a multi-platform installation, this release has been developed using version **2.4.5** of the GNU C compiler running under SunOS version **4.1.3** on a Sun iPC Sparc-based work-station. However little difficulty should be experienced in using any other ANSI standard C compilers. The source-level debugging support requires the GNU debugger and has been tested with **gdb** version **4.8**.

The components of this release produced at the University of Southampton are delivered with our standard license:

This software and the accompanying documentation are licenced free of charge with absolutely no warranty. Copyright of the software and documentation is retained by the University of Southampton.

Further copies of this software system may be made and distributed at will, provided that it is distributed in full with no modifications and that no charge is made other than media costs. Parties wishing to distribute modified versions of the package, including commercial products utilising components of the package, should contact the University for further clarification. Any derivative work must retain an acknowledgement to the University of Southampton and ESPRIT.

Since this software is licenced free of charge it is supplied without warranty of any kind, either expressed or implied. The entire risk as to the quality and performance of this software is with the user.

Production of the system has been funded by ESPRIT Project P5404, General Purpose MIMD.

2 Installation Procedure

Before installation you should ensure that the GNU C compiler and its associated utilities are installed on your system. After unarchiving the package the following changes to the shell environment variables need to be made:

- **\$SPOC** should be created and directed towards the base directory of the SPOC installation.
- **\$PATH** should contain **\$SPOC/bin/\$ARCH**, where **\$ARCH** indicates the host machine architecture. The distribution contains only a **sun4** setup consisting of **occ2c** and **omakef** binaries and a shell script for the debugger (**odebug**). SPARC binaries are also supplied in **\$SPOC/lib/sun4** for the following occam libraries: **dblmath**, **Intrinsics**, **convert** and **snglmath**.
- **\$OSEARCH** may be created to contain a space separated list of directories to search for occam includes and libraries. *All directory names should be slash terminated..* If the library or include file is not found on this path (or the environment variable has not been defined) the directory **\$SPOC/libs/** is checked. No **\$OSEARCH** variable is therefore required if the only libraries you are using are those supplied with the release.

3 A Simple Example

To demonstrate that the system has been installed correctly you are advised to try out this simple example. Switch to the directory **\$SPOC/examples/hello** and type:

```
omakef hello
```

this should generate a makefile called **hello.mkf**. An executable can now be generated and run by typing:

```
make -f hello.mkf
hello
```

Any options supplied to **omakef** will be passed through to the translator, so if you would like to look at this example using the debugger (not very interesting) you need only type:

```
make clean -f hello.mkf
omakef hello -g
make -f hello.mkf
odebug hello
```

At the **spod** prompt type:

```
step      ...a couple of times.  
quit
```

Other translator options are displayed by typing `occ2c -h`. Much more information about compiling and debugging `occam` programs using SPOC can be found in the User Guide.

4 Recompilation

The `occam` to C translator is composed of handwritten ANSI C header and code modules plus input files to the GMD Compiler Toolkit (which themselves contain fragments of C). The code is dependent upon the several features of the GNU C compiler and its tools. The most important of these is that all the `occam` debugging support is dependent upon use of the GNU Debugger. The others are that some of the primitive scheduler operations are written as `inline` functions to optimise their execution speed, and that the translation system makes use of the compiler's 64-bit integer support (the `long long` primitive type), as does the generated code. However these last two options are only used if the macro `_GNUC_` has been defined, its absence will result in slower code and no support for `occam INT64s`.

Source for the installation is distributed in the following directories.

- `$SPOC/spoc/source/gmd` - These files are processed by the the GMD compiler tools to generate C header and code files.
- `$SPOC/spoc/source/c` - Handwritten C header and code modules for the system.
- `$SPOC/spoc/source/tables` - Tables of keywords and additional tokens. These are processed to produce both input files to the GMD compiler tools and C header files.
- `$SPOC/spoc/build` - The C header and code files in this directory are those generated by the GMD Tools.

The system `Makefile` is distributed in the `build` directory and is setup to re-compile the system from the generated source code. Full recompilation of the system requires prior installation of the GMD Tools and the uncommenting of the line `include Gmd.mak` in the file `Occam2C.mak`. You should also set the Make variable `GMDTOOLS` to the base of your Compiler Toolkit installation. The GMD compiler tools are available via anonymous ftp from `ftp.gmd.de` (129.26.8.90) in the directory `gmd/cocktail`. You are advised to follow the installation procedure for these tools extremely carefully. The system requires a lot of disk space, if all you wish to do is recompile this system then you need only install the following components: `bnf,lalr,cg,common,puma` and `reuse`.

4.1 Recompiling for other architectures

Porting of the system to another architecture should be easy. Switch to the `build` directory and select an appropriate directory name for the new binaries, for example `sun3`, `solaris` or `linux`, and then make the new binaries by typing:

```
make select ARCH=newarch
```

Further `makes` (with no arguments) will target the newly selected architecture, to revert back to `sun4` compilation just type: `make select`

If the make fails to find an architecture dependent source file it will use the `sun4` versions as the default: creating new architecture directories and copying the files across as necessary. Depending upon your specific environment you may discover that minor changes need to be made to the new files. For example, you may find it necessary to override some of the standard environment variables such as `$CC` and `$CFLAGS`. The completed make should have generated new `omakef` and `occ2c` binaries in the appropriate directory off `bin`.

Modify your `$PATH` environment variable to locate the new binaries instead of the `sun4` ones and ensure that the `$SPOC` environment variable is directed towards the base directory of the installation.

You are reminded of the system's dependence on specific GNU C features described in Section 4.

4.2 Recompiling the distributed occam libraries

Once new binaries have been generated and the environment variables have been set up, recompilation of the occam libraries should be straightforward. Switch to the new architecture directory off `lib` then run `libinstall`, followed by `make`. For this to work it is vital that the new `occ2c` and `omakef` binaries are on the path and that the `SPOC` environment variable has been set.

If you need to alter the makefiles generated by `omakef` (for example if no `ranlib` is required) then the file `$SPOC/lib/$(ARCH)/Omakef.tpl` can be edited. Typically it should be sufficient to modify the header part of the template file, as this declares many standard make macros which are applied by the other template sections.

4.3 Compiling the proprietary Inmos libraries

If you have access to the sources of the Inmos libraries then full compatibility can be obtained for `hostio.lib`, `string.lib` and `streamio.lib`. Generation of new binaries is performed as follows:

1. Set environment variable `INMOS_SRC` to point to the directory containing the Inmos library sources.
2. Switch to the appropriate architecture directory (off `$SPOC/lib`) and run `libinstall`. If you have already installed the libraries in this directory, you should remove `Makefile` and the `hostio`, `string` directories, before re-running the `libinstall` script.
3. Run `make` to compile the libraries

The `imsmake.tpl` file is prepended automatically to the `Omakef.tpl` file and has the following modifications: the `-s` option has been added to the top-level occam compilation, and the `spserver` library has been added to those that must be linked in to the application binary.

The Inmos libraries will work on both little-endian and big-endian architectures, but a compile-time `#define` must be correctly set. It is recommended that `$SPOC/lib/src/spserver/inmos.h` should be checked and possibly modified, before compilation of any libraries. The endian-ness of both the target machine and the `spserver` protocol transactions must be correctly defined.

There are four distinct possibilities.

Target	Protocol	#define	occ2c options
big	big	BIG_AND_BIG	
big	little	BIG_AND_LITTLE	-mi -me
little	little	LITTLE_AND_LITTLE	
little	big	LITTLE_AND_BIG	-mi -me

Note: The `occ2c` options in the above table should be added / removed from the `Omakef.tpl` file in `$SPOC/lib/$(ARCH)` directory, before **ANY** of the libraries are compiled, and **should not** be modified without recompilation of the libraries.

5 Installing the manual pages

To install the manual pages select a destination directory into which to install them, for example `/usr/local/man` (the default is `$SPOC/man`). Make sure that the directory exists and

that you have the appropriate priviledges to write to it. Switch to `man/source` and type `make MANDIR=/usr/local/man`. If the directory is not already on the `MANPATH` then add this directory to it.

We would like to thank David Beckett at Kent for allowing us to distribute his manual pages for the standard occam libraries which are integrated into this release.

6 Bug Reports

Any problems with this system should be reported to Dr Denis Nicole `dan@uk.ac.soton.ecs` quoting the version number and compilation date of the Occam 2 C translator (given by the `-h` option).

If the translator fails to parse your occam or generates the wrong code please reduce the code to the smallest example which demonstrates the fault before sending us your sources. Please report any bugs as soon as possible.