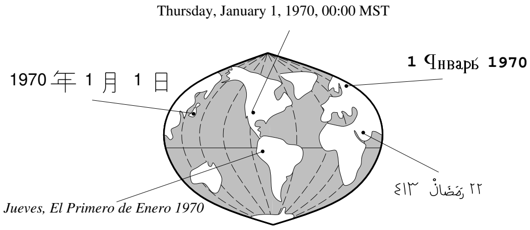
The MultiCal Project



Installation Instructions for MultiCal

Curtis E. Dyreson

October 29, 1993

Release 1.0

Abstract

These are the installation instructions for MultiCal. By following these instructions, installation should be quick and easy.

The MultiCal Project
Department of Computer Science
The University of Arizona
Tucson, AZ 85721
multical@cs.arizona.edu

Title

Installation Instructions for MultiCal

Author(s)

Curtis E. Dyreson

Publication History

October 1993. Version 1.0.

Copyright and trademarks

The MultiCal Project
Department of Computer Science
The University of Arizona
Tucson, AZ 85721
multical@cs.arizona.edu

Contents

1 Installation Instructions for MultiCal

This document explains how to install MultiCal, and lists the resources needed to build and maintain the system. We strongly suggest that you read the entire document before proceeding with the installation.

2 Required Resources

2.1 Software Requirements

To build MultiCal, you will need a compiler that supports 64 bit integers (`long long`). The Gnu C compiler (`gcc`), version 2.0 or above, is one such compiler, and MultiCal can be compiled with the appropriate version of `gcc`.

2.2 Disk Space Requirements

A tarred and compressed file, `multical0.9.tar.Z`, contains the entire system and occupies about 3MB. Uncompressed, the system is about 9MB of which 8MB is source code. The required object and source code for the system will occupy approximately 17MB of disk space after building. These figures assume that the system has been compiled in optimize mode (see below). If the system is compiled in debug mode, the size of the executables and the library will increase substantially, perhaps by a factor of 2.

2.3 Main Memory

The installation should be possible in 8MB of main memory, though 16MB is certainly recommended. All of our development has been on machines with at least 16MB of main memory.

2.4 Execution Time

The entire installation takes between 20 and 30 minutes, depending primarily on the type of machine and file server. The break-down by `make` options is shown in Table 1. All times are in minutes, and assume an otherwise unloaded machine (SUN-4 IPC) and lightly loaded file server. A `make install` builds the entire system and installs the executables in the appropriate directory. A `make system` also builds the system (save for the executables). A `make clean` removes all object files (and the executables). Finally, a `make checkout` tests the executables to determine if there were problems in the installation. Currently `make test` is unsupported.

3 Site Dependencies

The next section explains the steps for installing MultiCal. These instructions vary depending on the machine and the operating system on which MultiCal is installed. Specifically, the following potential problems are addressed in the installation procedure.

- There are bugs in `yacc` on most machines; it either core dumps or produces bad code. Therefore, we include the generated C files in the distribution (generated on a Sparc architecture) and skip running `yacc` by commenting out the `HASYACC` macro.

Step	*Approximate Execution Time* (in minutes)
install	25
system	20
clean	4
test	15
checkout	1

Table 1: Installation Times

- The scorpion system and, in particular, `idlc` is not supported everywhere. Consequently we include the generated IDL files in the distribution (generated on a Sparc architecture) and skip running `idlc` by commenting out the `IDLC` macro.

4 Installation Instructions

To install MultiCal, perform the following steps in order. We suggest that you read through this entire document before performing any of the steps, to anticipate problems before they occur.

1. Change your current working directory to the top-level MultiCal directory. This is where MultiCal will reside. We will call this directory `root/`. Copy the file `multical0.9.tar.Z` into `root/` from the FTP area. Uncompress the file by running:

   ```
   uncompress multical0.9.tar.Z
   ```

   The file `multical0.9.tar` will be created when `uncompress` successfully terminates.

2. MultiCal is extracted by untarring the file created above. This may be done as follows and will take several minutes.

   ```
   tar xf multical0.9.tar
   ```

   A directory `multical0.9` will be created in `root/`.

3. To reduce disk space the old tar file may now be deleted:

   ```
   rm multical0.9.tar
   ```

4. Now change directory to the multical directory:

   ```
   cd multical0.9
   ```

   This directory contains a file named `Makefile` and one named `Makefile.include`, as well as other files and directories. Examine the file `NOTES`; this discusses important details on the release not covered by the installation instructions. Also examine the file `README`.

5. At this point, it is necessary to customize the file `Makefile.include`. `Makefile.include` will be included into other system `Makefile`s and contains path names for Unix utilities used throughout the makefiles (which may be site specific) as well as all the parameters for controlling the compilation of MultiCal. The first and most important edit is to the line reading

```
TOPDIR=/usr/rts/...
```

You should edit this line to point to the current directory, e.g.,

```
TOPDIR=/usr/me/multical0.9
```

Another important customization is to edit the line

```
ARCHVAR=sparc
```

to match the architecture of your system. To specify different paths or options for the UNIX utilities edit the following lines in `Makefile.include`. Please check the following variables very carefully, because they are used extensively in the installation. Every site will most likely have to change at least some of these variables. For `gcc` compilers below version 2.0, support for long longs is a "hidden feature." Multical may be compiled (or maybe not!) with such versions but it will probably be necessary to include a macro definition, e.g.,

```
CC=gcc -Dalloca=malloc
```

as many compilers do not know how to correctly generate code for alloca.

```
CHMOD=/bin/chmod
CHOWN=/etc/chown bin
CHGRP=/bin/chgrp multical
LD=ld
AR=/bin/ar ru
CAT=/bin/cat
CD=cd
CC=gcc
CP=/bin/cp
FIND=/usr/bin/find
LINT=/usr/bin/lint
MAKE=/bin/make
MKDIR=/bin/mkdir
MV=/bin/mv
PC=/usr/lang/pc
RANLIB=/usr/bin/ranlib
RM=/bin/rm
RMDIR=/bin/rmdir
SED=/bin/sed
SHELL=/bin/sh
STRIP=/bin/strip
TOUCH=/bin/touch -c
#OURLEX=/usr/bin/lex
#OURYACC=/usr/local/bison -d -v
```

```
#IDLC=/usr/local/scorpion/bin/idlc
UMASK=umask
LN=ln
CSHELL=/usr/bin/csh
LPR=lpr
```

If some paths are unknown, the appropriate path for each utility can be determined by executing

```
bin/wherearethey
```

If any of the above is not needed or cannot be used set it to `$(NOOP)`, a command that does nothing. As an example, if you don't want to change the privileges, edit the `CHMOD` variable to:

```
CHMOD=$(NOOP)
```

The entire system may be built in `OPTIMIZE` or `DEBUG` mode. The `HOWTOCOMPILE` variable may be modified as needed. In `DEBUG` mode additional symbol table information is stored for each tool in the system. This requires extra disk space and main memory, but allows the tools to be debugged later. Currently, this variable is set to optimize mode.

```
HOWTOCOMPILE=OPTIMIZE
```

At this point, `Makefile.include` has been customized.

6. To verify that the path settings of the UNIX tools are correct, type

   ```
   make testpath
   ```

   If any serious errors occur, recheck the paths before continuing. This step may be repeated as needed.

7. To build MULTICAL type

   ```
   make install >& make.all.log &
   ```

   Throughout the installation, a very few minor warning messages will appear, and can be ignored. However, any error generated by `make`, indicating a failed make command, are significant, and should be addressed. If any such errors are encountered, the step containing the error or `make install` can usually be redone after correcting the error, as many times as necessary. After the installation is complete, if any programs or libraries need to be recompiled, consult the individual `Makefile` for the proper target to use in the `make` command.

8. If there are problems, check with the MULTICAL developers at `multical@cs.arizona.edu`.

9. The installation is complete; savor the moment!

10. Now, you may test MULTICAL by typing

    ```
    make checkout
    ```

    You should see a message indicating that MULTICAL successfully ran.

11. All the documentation for the system has been included on-line as postscript documents. To print out the entire documentation (approximately 300 pages), do the following:

    ```
    make printdocs
    ```

    If your printer programs differ substantially from the default ones, you may need to edit `Makefile.include` to customize the print commands.

12. You may wish to start MultiCal once the installation is complete.

    ```
    cd prototype/test
    ../bin/multical -v example.sql
    ../bin/multical -v example.bad
    ```

5 Acknowledgements

These instructions were derived from those provided with the Scorpion system, also developed at the University of Arizona. Curtis Dyreson prepared release 0.9.

This work was supported in part by NSF grants ISI-8902707 and IRI-9302244, and in part by IBM contract #1124.