

within the declared array bounds.

-fast Select the combination of compilation options that optimizes for speed of execution without excessive compilation time. This should provide close to the maximum performance for most realistic applications.

Diagnostics

The diagnostics produced by f77 itself are intended to be self-explanatory. Occasional messages may be produced by the linker. The error messages are listed in perror(3f).

PC Pascal compiler

PC or pc -g "pascal source code file ending in .p" will produce an executable file "a.out" upon successful compilation.

Prolog

Quintus Prolog is installed on workstations: atrium51, acc1 and acc3
rsh (rlogin) to one of the above servers and enter " prolog".
Quintus Prolog uses an emacs based window and has a built in help system.

ML

The functional language ml is installed in /users7/ml. The ml documentation is available under /users7/ml/doc, mainly as post-script files. Use pageview or ghostview to view on line.

Scheme

The scheme programming language is installed in /users7/scheme, and maybe started by bchscheme or scheme . The scheme documentation is available under /users7/scheme/doc.

Perl

"perl "

Java

"javac file.java" to compile

"java file " to run

Python

"python options script "

GCC

gcc

g++

The C and C++ compilers are integrated. Both process input files through one or more of four stages: preprocessing, compilation, assembly, and linking. Source filename suffixes identify the source language, but which name you use for the compiler governs default assumptions:

gcc assumes preprocessed (.i) files are C and assumes C style linking.

g++ assumes preprocessed (.i) files are C++ and assumes C++ style linking.

Suffixes of source file names indicate the language and kind of processing to be done:

.c C source; preprocess, compile, assemble
.C C++ source; preprocess, compile, assemble
.cc C++ source; preprocess, compile, assemble
.cxx C++ source; preprocess, compile, assemble
.m Objective-C source; preprocess, compile, assemble
.i preprocessed C; compile, assemble
.ii preprocessed C++; compile, assemble
.s Assembler source; assemble
.S Assembler source; preprocess, assemble
.h Preprocessor file; not usually named on command line
?? Other (unrecognized) files passed to linker.

Common cases:

.o Object file
.a Archive file

Linking is always the last stage unless you use one of the -c, -S, or -E options to avoid it (or unless compilation errors stop the whole process). For the link stage, all .o files corresponding to source files, -l libraries, unrecognized filenames (including named .o object files and .a archives) are passed to the linker in command-line order.

OPTIONS Options must be separate: '-dr' is quite different from '-d -r'.

Most '-f' and '-W' options have two contrary forms: -fname and -fno-name (or -Wname and -Wno-name).

A copy of the manuals "Using and Porting GNU GCC" and "User Guide to the GNU C++ Library" are available at the front desk of the Atrium Lab.

Debuggers

First compile your code with the `-g[n]` option to insert debugging code.

dbx

Debuggers best suited for ACC and C++ compiled programs
debugger

`dbxtool` [executable] [core]

`debugger` [executable] [core]

`dbx` [executable] [core]

`dbx` is a utility for source-level debugging and execution of programs written in C++, ANSI C, FORTRAN, and Pascal. `dbx` accepts the same commands as `dbxtool(1)` & `debugger(1)`, but uses a standard terminal (tty) interface. For a full overview of `dbx` and Debugger features see the Debugging Tools user guide.

Debuggers best suited for gcc and g++ compiled programs

gdb

xxgdb

ddd

`gdb` [executable] [core]

`xxgdb` [executable] [core]

`ddd` [executable] [core]