

**GNU-C.html**

**COLLABORATORS**

	<i>TITLE :</i> GNU-C.html		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 2, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>GNU-C.html</b>	<b>1</b>
1.1	GNU-C.html . . . . .	1

# Chapter 1

## GNU-C.html

### 1.1 GNU-C.html

bUsing GNU-Cub

I'm currently using GNU C for my C++ programming, so I'd thought I'd write a little bit about it, as well giving a few hints to solving possible problems.

I got hold of it from CUCD6, and I still use it from there (a hard disk installation takes up quite a bit of space). It is available in a load of lha archives from Aminet and most PD libraries too.

When I first looked at it, it seemed a nightmare; loads of different documents, not really being of any help, and the bin drawer contained all sorts of programs which might have been the compiler... So this is what you do:

Open a shell window - you'll need this for your C working, so using something like ZShell or KingCON is best. Now with this diskmag, in the Mag4/ drawer should be a little file called 'GNU-C\_Start'. You need to edit it (use ed, or something), to change it to your needs. There are comments in it telling you what to do - the important bit is assigning GNU: to the correct place. Now run this script by typing:

```
execute ../gnu-c_start
```

Where '../' is the required pathname. (NB, this is why one of the aforementioned shell replacements is advisable, as you can just drag the script icon into the shell window. Also note that although there is an icon for this script, do NOT run it with IconX. It must be run from the shell you are working from!)

Now to compile a program, use the format:

```
gccv -o [outputfile] [source code file]
```

You might like to use the cd command to move to the directory storing your C work, or do this in the script. Try a good old Hello World program if you need an example:

```
// Hello World!  
#include <stdio.h>  
  
void main()  
{  
printf("Hello DoubleClick Readers!\n");  
}
```

Save it as 'Hello.c', and compile from the shell with:

```
gccv -o Hello Hello.c
```

And run with:

---

Hello

Remember! You must be in the right directory in the shell, or specify the directory in the file pathnames.

So, you want to do some C++ programming? GNU C will do this too, and it looks at the extension of your source file to tell if it is C or C++. Normal C files have '.c' on the end. By typing '.C' [ie, capital C] GNU will assume the file is in C++. Remember of course that AmigaDOS does not care about letter cases in filenames, so you can save as '.c', just remember to type '.C' when compiling! NB, other extensions can be used to indicate C++ to (like '.CPP', I think).

Mark Harman