Compiling C modules

Table of Contents

modules in C

C programming projects in industry have more than one source file because it usually takes more than one programmer to complete the job. However, only one of those C source files in an application contains `int main() { }`. Other C source files are known conceptually as modules. A module's source code contains functions() which work independently or together with other modules. A "main" program calls those functions.

Source files making up an application are grouped together in a Visual Studio IDE Project or in the same folder/workspace when using Visual Studio Code or other development tools including command line compilation.

A typical C application has .h header files, .c module files, and a single main.c source file which calls functions in the modules.

macOS

- Visual Studio Code or Xcode are good choices for C development.
- The gcc command line compiler is available.

Windows

- Visual Studio IDE is the professional's choice for C development on Windows.
- Visual Studio Code will also work.

- Minimalist GNU for Windows project (MinGW-w64) has a port of the GNU Compiler Collection (gcc) providing "A complete runtime environment for GCC & LLVM for 32 and 64 bit Windows" MinGW-W64 is an up-to-date project that is in active development. The original MinGW.org Project website is defunct as of April 2021.
    - **To install and use, see https://winlibs.com/**
- The gcc compiler is native to the Unix / Linux world. If you are going all hardcore, you can do it in the Windows Subsystem for Linux (WSL) where gcc is very happy.
    1. Open PowerShell terminal window to install WSL
       `> wsl --install` *[installs the recommended Ubuntu distro]*
    2. Start WSL Bash shell in default directory
       `> wsl ~`
    3. Update and upgrade WSL (to be safe):
       `# sudo apt-get update && sudo apt-get upgrade -y`
    4. Clean unrequired packages:
       `# sudo apt autoremove -y`
    5. Install GCC:
       `# sudo apt-get install gcc -y`
    6. Check and confirmed installed gcc version:
       `# gcc –version`
       `gcc (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0`

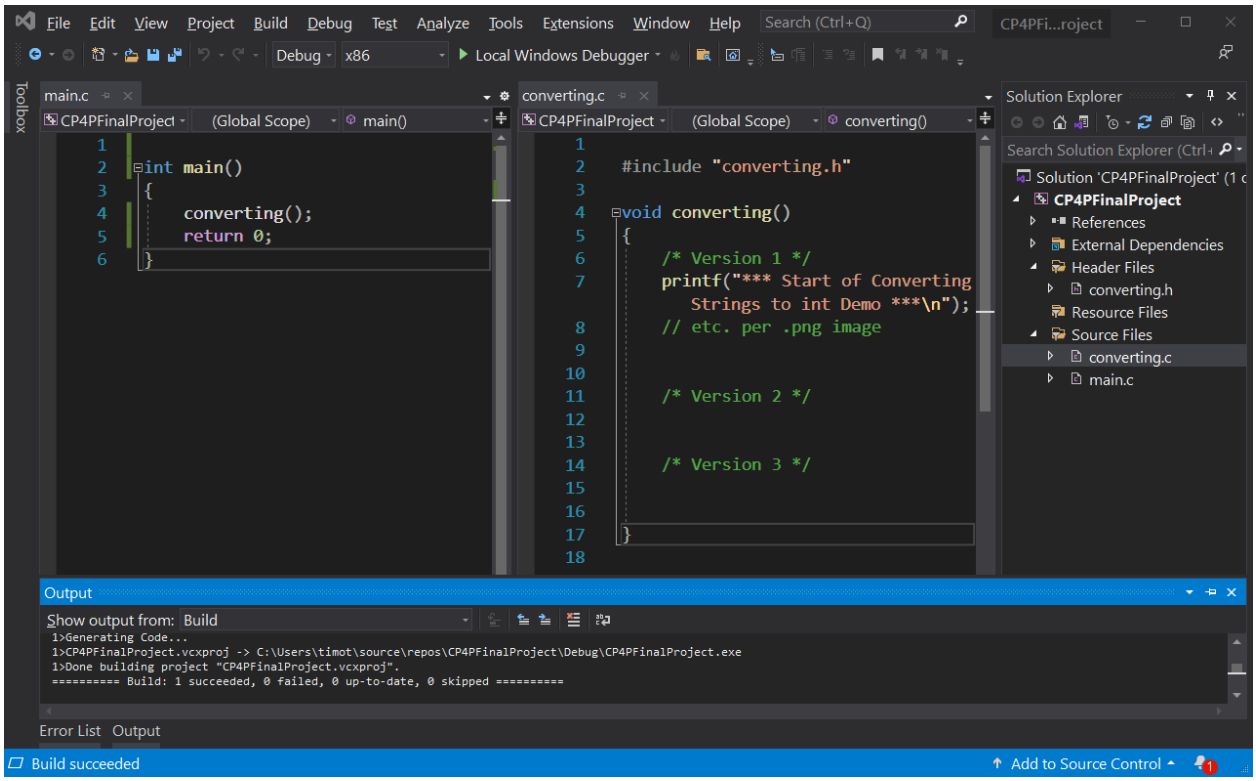
helloWorld C source to test a compilation

```
#define _CRT_SECURE_NO_WARNINGS // add to top before any #include re use of strcpy()
/*
helloWorld : the canonical test of any programming language thanks to K&R.
*/
#include <stdio.h>  // Standard Input/Output
int main(void)                  // mainline – only one in an application
{
    // console output as proof of compiler installation and operation

  // call a function in the standard input/output library.
    printf("Hello, World!\nThis is a compiler test.\n");

    return 0;
}
```

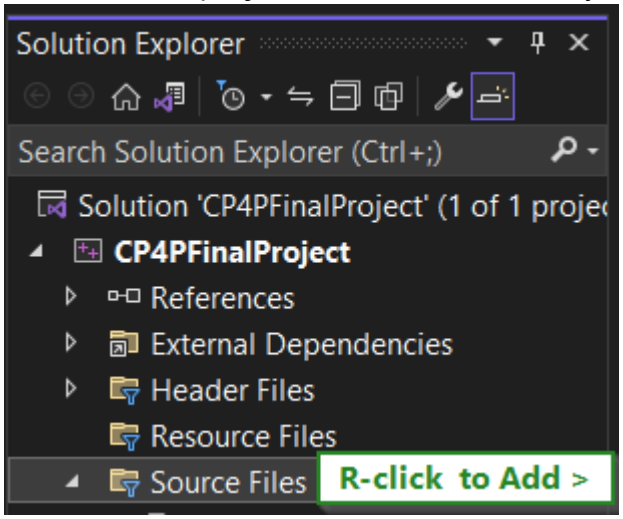
project C source template example

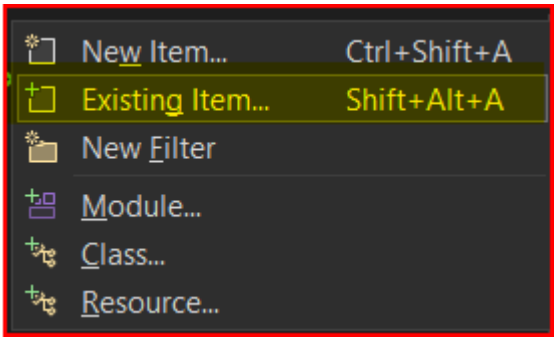


Your project or workspace/folder/directory contains three files:

- *moduleName*.h header file
- *moduleName*.c function file
- main.c with int main() { which calls the function inside *moduleName*.c }

N.B. Project Leader:
When incorporating modules created by others, those "Existing Item" source files **must be added** to the project with main.c else they will not be included in the Build (compile) process.





gcc compilation of Final Project source files

macOS or Linux or WSL

To compile only a module for unit testing and make it runnable:

```
   > gcc -nostartfiles      module.c  -o module
```

e.g. `gcc -nostartfiles  converting.c  -o converting`

To compile a module for unit testing with a main() caller:

`> gcc  moduleName.c  main.c  -o main`

e.g.

```
> gcc  converting.c  main.c  -o main
```

To compile all modules into a program for Integration testing,
specify all the module names:

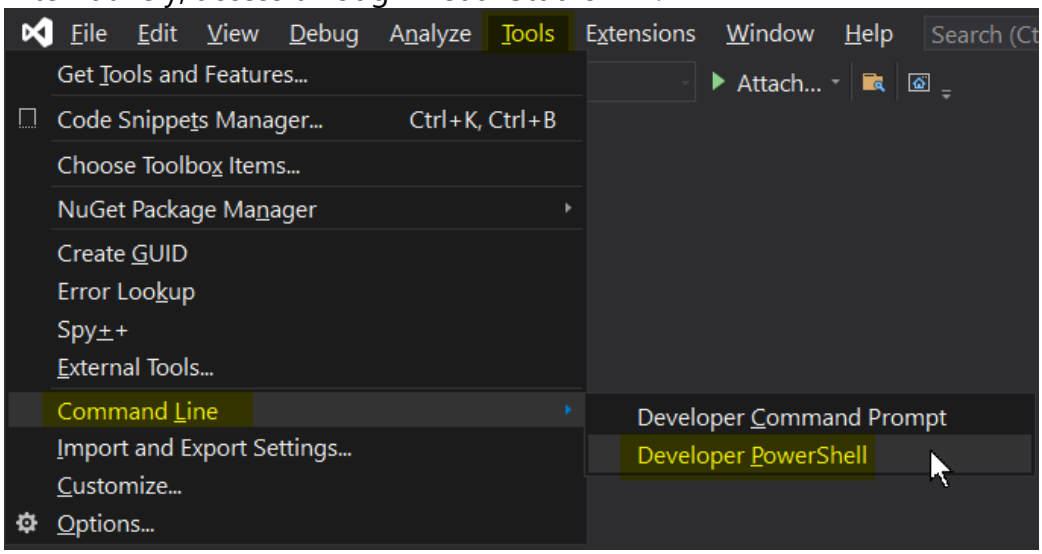
```
gcc moduleA.c moduleB.c moduleC.c moduleD.c main.c -o main
```

e.g.

```
gcc fundamentals.c manipulating.c converting.c tokenizing.c  main.c -o main
```

Microsoft cl compilation of Final Project source files

The **cl** compiler runs only from a Visual Studio developer command prompt.
VS-IDE or VS Code > menu > View > Terminal [Ctrl + `] shows the terminal.
Alternatively, access through Visual Studio IDE:



****
** Visual Studio 2022 Developer PowerShell v17.3.5
****
PS C:\Users\me\source\repos\CP4PFinalProject>

---

cd "C:\Users*me**Documents**Seneca**CPR101**Final*" —> as required

PS C:\Users\me\ Documents\Seneca\CPR101\Final>

cl .*moduleName*.c .\main.c /link /out:main.exe

cl .\converting.c .\main.c /link /out:main.exe

cl .\converting.c ### source requires main() to call *function*()

Microsoft (R) C/C++ Optimizing Compiler Version 19.29.30136 for x86
Copyright (C) Microsoft Corporation. All rights reserved.

converting.c
main.c
Generating Code...
Microsoft (R) Incremental Linker Version 14.29.30136.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:converting.exe
/out:main.exe
converting.obj
main.obj
PS C:\Users\timot\source\repos\CP4PFinalProject> .\main.exe
*** Start of Converting Strings to int Demo ***
...

Windows Security

You will likely have to allow the compiler permission to write an .exe to your working folder, especially if you see this message:

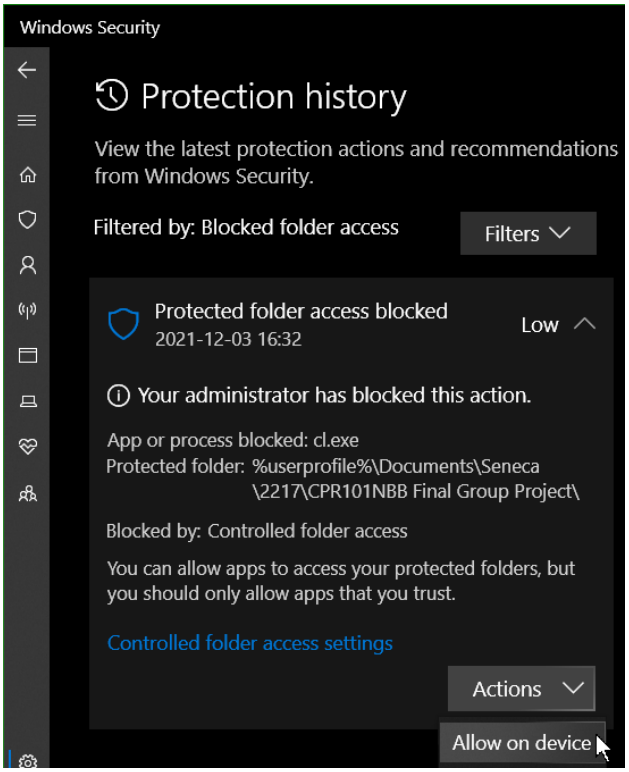
.../bin/ld.exe: final link failed: No space left on device

There is "no space" because Windows Defender denied it access. The fix is to...

Windows-key > "Ransomware protection" > Block history > click the latest item > Actions > Allow on device (see screen shots below)
https://geekthis.net/post/mingw-fix-permission-denied-ld-and-error/

One time only:


Windows Security
Protection history
View the latest protection actions and recommendations from Windows Security.
Filtered by: Blocked folder access
Filters
Protected folder access blocked
2021-12-03 16:32
Low


Windows Security
Protection history
View the latest protection actions and recommendations from Windows Security.
Filtered by: Blocked folder access
Filters
Protected folder access blocked
2021-12-03 16:32
Low
Your administrator has blocked this action.
App or process blocked: cl.exe
Protected folder: %userprofile%\Documents\Seneca \2217\CPR101NBB Final Group Project\
Blocked by: Controlled folder access
You can allow apps to access your protected folders, but you should only allow apps that you trust.
Controlled folder access settings
Actions
Allow on device

PowerShell compilation script

To run a PowerShell script [below] which will make compilation easier,
right click the compile.ps1 file and "Run with PowerShell".

The script will prompt to enter the C source path & filename.

Alternatively, open a PowerShell terminal/console in your C files folder, type c and press [TAB] until "compile.ps1" appears, space bar, then the first letter of the source filename and press [TAB] until the desired .c file appears. Press Enter.

Start Notepad and save the following as `compile.ps1`

```
# PS script to compile and run a C program or module.
    Param (
    [Parameter(Position = 0, Mandatory=$True)]
    [ValidateNotNull()]
    $source_file)

Function EndOfJob()
{
    # restore environment PATH
    Set-Item -Path Env:Path -Value $originalPath
    Read-Host -Prompt "`n> End of program. Press ENTER to continue."
    exit
}

# temporarily add MinGW folder to environment PATH
$originalPath = $Env:Path
Set-Item -Path Env:Path -Value ("C:\Program Files (x86)\mingw-
w64\mingw32\bin;" + $Env:Path )
# the above path may require tweaking for your local PC

if ($source_file -eq "")
{
    Write-Host "*** no source file input ***"
    EndOfJob
}
elseif (-not (Test-Path -Path $source_file))
{
    Write-Host "source file not found: " $source_file
    EndOfJob
}

Write-Host "`nCompile C source file" $source_file
```

```
$source_file_name = (Get-Item $source_file ).Basename # file name without
extension

# compile source file as source filename(.exe)

if (Select-String -Path $source_file -Pattern "main(" -SimpleMatch -Quiet)
{
    # source.c contains main()
    gcc $source_file -o $source_file_name
}
else
{
    # -nostartfiles switch allows compilation without main()
    gcc -nostartfiles $source_file -o $source_file_name
}


if ($LastExitCode -ne 0)
{
    echo "See above compilation related error."
}

if (-not (Test-Path -Path ($source_file_name + ".exe")))
{
    Write-Host "Executable not found for " $source_file_name "`nCheck
Security / Protected folder access blocked for 'ld.exe' or 'as.exe'`nor a
source code compile error."
    EndOfJob
}

Write-Host "> Running" $source_file_name "`n"
& .\$source_file_name

EndOfJob
```