

gcc compiler for your PC

Apple mac users likely already have a gcc compiler if Xcode or Visual Studio Code is installed.

Microsoft cl compiler: <https://youtu.be/rqLbyj0Tnlg>

gcc compiler: <https://youtu.be/xvs7oQZG3jI>

gcc compiler on Mac: <https://youtu.be/we2Oc4WQ7FM>

Minimalist GNU for Windows (MinGW) has a port of the GNU Compiler Collection ( gcc ).

**Download the [MinGW-W64 Online Installer](#)** (provenance noted below)

Run mingw-w64-install.exe

Override the installation folder to "C:\Program Files (x86)\mingw-w64"  
i.e. without the version sub-folder.

**To start the MinGW compiler environment**, Windows-key > "mingw-w64" which appears as "Run Terminal". The menu item is "MinGW-W64 project". At the prompt, verify install with > gcc -version

To compile a module for [unit testing](#),

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

e.g. > gcc -nostartfiles converting1.c -o converting

To compile all modules into a program for [Integration \(~system\) testing](#),

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

e.g. > gcc fundamentals1.c manipulating1.c converting1.c tokenizing1.c  
main.c -o main

A PowerShell script to compile and run a module or program is documented below.

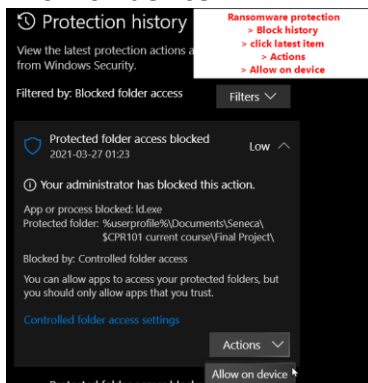
## Troubleshooting

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



<https://geekthis.net/post/mingw-fix-permission-denied-ld-and-error/>

### PowerShell compilation script

Start Notepad and save the following as `compile.ps1` or whatever name you like so long as it ends in `".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 )

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
```

To run the script, right click the `compile.ps1` file and Run with PowerShell. Then enter the source filename. Alternatively, open a PowerShell console in your `.c` files folder, type `c` and

## gcc compiler for your PC

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.

A test C program:

```
/*  
helloWorld : the canonical test of any programming language thanks to K&R.  
*/  
#include <stdio.h>          // Standard Input/Output  
int main(void)              // mainline  
{  
    // console output as proof of compiler installation and operation  
    printf("Hello, World!\n");  
    printf("This is a test of the MinGW gcc compiler.\n");  
    return 0;  
}
```

MinGW-W64 appears to be a more up-to-date project than the original MinGW32.

The MinGW-W64 Online Installer came from:

<http://mingw-w64.org/> From their Downloads page, click [MingW-W64-builds](https://sourceforge.net/projects/mingw-w64/) which leads to <https://sourceforge.net/projects/mingw-w64/> That landing page Download is a .zip file that contains a library of source code only.

Go to Files, and scroll down to MinGW-W64 Online Installer.



## MinGW-W64 Online Installer

- [MinGW-W64-install.exe](#)

Please ignore the idiotic comments seen there.

The original MinGW.org Project website is defunct as of April 2021. There is a page at <https://mingw.osdn.io/> but it references the .org site and cautions against using anything found at SourceForge. The [MinGW32](#) compiler is available from <https://osdn.net/projects/mingw/> There is a multi-step process; first install the MinGW Installation Manager.