ATAD | Software, Alternative Methods

These options should be used if you have exausted the methods described in the Software.pdf document, namely for Windows and, particularly, MacOS.

They are presented in preferred order:

1. Virtual Machine (works on all OS environments), or;
2. VS Code and MinGW (only for school workstations).

1 | Virtual Machine

Using a virtual machine will give you a virtualized Linux operating system (guest) on your main operating system (host).

This solution may be heavier on resources than the previous method.

1. Follow the instructions from here.
2. Follow the instructions from Software.pdf, starting at step 6 of "Manual installation (Windows/WSL or Linux)".

> The **Shared Folders** functionality is higly advised, allowing you to keep all your projects in your main operating system (host) filesystem.

2 | MinGW + VS Code Extension

This is the last option and you'll be left with an incomplete *toolchain*. Namely, **you'll be left without**:

- *Valgrind* (memory checker);
- Possibly, *Doxygen* (documentation), depending on the MinGW installation.

However, **it will be the option reserved for the school workstations to perform the assignments**, if you don't have a personal laptop.

> The MinGW installation can also be performed on a personal USB thumb drive and all development done from it.

MinGW VS Code Extension

To successfully use MinGW you'll need to install the following VS Code extension:

- MinGW C Configuration

For each opened project, run the command provided by the extension to configure the VS Code project. Follow the instructions on the extension's page.

Installation of MinGW on a USB Drive

> Check if the school workstations has MinGW installed, i.e., check for the existence of a MinGW folder (`C:\MinGW` or `C:\mingw32`). If so, you may use this installation.

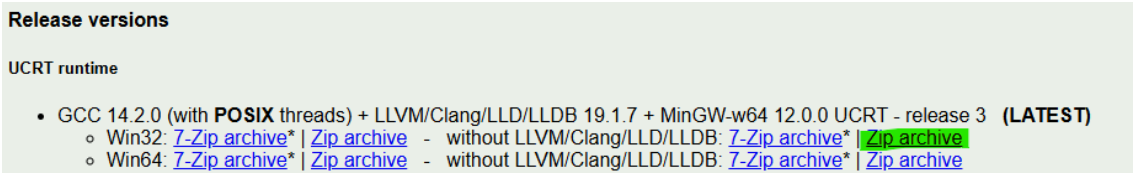
Useful Links:

- WinLibs standalone build of GCC: Link

Perform the following steps, where `<DRIVE>` means the installation drive, e.g., `D:` (pen drive):

1. Follow the *WinLibs* link above.

   Look for the latest **Win32 - without LLVM/Clang/LLD/LLDB** *zip* file, e.g.:

   
   Release versions

   UCRT runtime

   - GCC 14.2.0 (with **POSIX** threads) + LLVM/Clang/LLD/LLDB 19.1.7 + MinGW-w64 12.0.0 UCRT - release 3 **(LATEST)**
     - Win32: 7-Zip archive* | Zip archive - without LLVM/Clang/LLD/LLDB: 7-Zip archive* | Zip archive
     - Win64: 7-Zip archive* | Zip archive - without LLVM/Clang/LLD/LLDB: 7-Zip archive* | Zip archive

   Save it to your computer.

2. Extract the folder `mingw32` from the *zip* file to `<DRIVE>:\`, i.e., you should be left with an installation folder like `D:\mingw32\`.

3. That's it! From now on it is important to use the provided *extension* which will inject the necessary configurations to find the *binaries* for this installation.

Author and support

Bruno Silva (bruno.silva@estsetubal.ips.pt)

You should ask your PL teacher for any help regarding these contents and procedures.