Linux Blockers

Things at DAE that prevent going full-Linux.

- [Respondus Lockdown Browser](#)
- [Adobe Photoshop](#)
- [MSVC Compiler Quirks](#)

Respondus Lockdown Browser

This US-based program is required to take certain exams, notably:

Semester 1:

- Algorithms
- Visual Language
- Graphics Fundamentals

Semester 2:

- Game Tech

Semester 3:

- Programming 3
- Graphics Programming (only the quiz)

Respondus has taken (some) measures to prevent Linux users from accessing it. The software attempts to detect when being run via WINE or a virtual machine.

Although its security is easily defeated, and Respondus itself encourages you to try it, please do **not** try to run it on Linux or a VM during an exam! The browser will **falsely** flag you as a cheater!

> Yes, you have our blessing...go ahead and see if you can break it
>
> -Respondus ([source](#))

And it goes without saying, we do not support any kind of cheating exams. Remember, we are here to learn: you'd not only be cheating the exam, but yourself.

Solutions

You can request a paper copy of the exam ahead of time, but it is likely you will be asked to use a Windows laptop.

Further reading

A university in Italy lost a lawsuit from its students over it using Lockdown Browser. The case cites concerns over the large amount of unnecessary data collection done by Respondus, and the unlawful (under GDPR) transfer of personal student data to the US. Note however that DAE does not employ the camera feature of the browser, so not all the points apply.

See this page from the Linux and Unix Users Group at Virginia Tech Wiki for more details on the browser: Lockdown Browser.

Adobe Photoshop

2D for Games is the only course with a hard requirement on Adobe Photoshop. This is because .psd files are required for submission.

Solutions

Photopea

You can use the excellent (but non-FOSS) web app Photopea as a nearly full-featured replacement for Photoshop.

Virtual machine

The regular distribution of Adobe Photoshop works fine on virtual machines (preferably with GPU passthrough, to enable all features). You can use a local VM (libvirt/QEMU works great), or even a cloud "virtual desktop" provider, to run Photoshop and activate it via your student email. If using a local VM, you can even run software like WinApps or Cassowary to seamlessly-ish integrate it into your desktop (NOTE: both are unmaintained, but still work).

Alternatives

Ask the teachers to use a FOSS alternatives such as Krita or GIMP! As of 2023, the final assignment only requires vector art, so Inkscape may be a good option as well.

WINE (not recommended)

Although it is possible to run recent-ish Photoshop versions on WINE quite well, Adobe's DRM (Digital Restrictions Management) system does not work. You can try alternative installation methods found on GitHub, but this is not recommended. If you choose to try this, please be aware that the account sign-in does not work so you should use a trial version.

MSVC Compiler Quirks

Visual Studio by default uses Microsoft's proprietary compiler MSVC, which often does not compile programs that work fine on Linux compilers such as `g++` or `clang++`. Therefore, assignments that require submitting source code should either find a way to make Visual Studio use an open-source compiler that can be tested on Linux without any change from teachers (if you do, please edit this page!), or make sure you test your code on MSVC!

Common causes

`windows.h` defines

Calls to functions whose names get clobbered by `#define`s in `windows.h` will likely break. Sometimes globally defining `WIN32_LEAN_AND_MEAN` and `NOMINMAX` helps, otherwise you will have to manually do this for all macros that affect your code:

```
#ifdef min
#undef min
#endif

#ifdef max
#undef max
#endif

// and so on...
```

`#include` extensions

MSVC allows backslashes in `#include` paths, which is nonstandard. You can pass `-fms-extensions` to clang to bypass this, but ideally `#include`s should be corrected.

MSVC also does not check the casing of filenames, as this is not relevant on Windows. When building on Linux however, this is very important. Make sure the path in the `#include` directive matches exactly that of the file, down to the casing!

Transitive includes

Some standard library headers may include other headers, such as `string` including `vector` implicitly. This is not defined by the standard (in fact, it is permitted!), so may break your code when switching compilers if you unknowingly depend on them.

The tool [include-what-you-use](#) can be used to solve this problem.

If using Clang, you can also reduce the number of these transitive includes by defining `_LIBCPP_REMOVE_TRANSITIVE_INCLUDES` (see the [docs](#) for more info).

C++ Standard extensions

Don't use extensions to the C++ standard, as these may not be available on MSVC. On modern CMake versions, you can set the target property `CXX_EXTENSIONS` to `OFF`.

C++ features

MSVC is often behind other compilers in implementing newer C++ features, so stick to older standard versions if you can. In older versions, MSVC usually has about the same support as GCC, and more than Clang, so you should be fine. You can see the current status on [cppreference.](#)

Solutions

MSVC on WINE

There are packages in the AUR that do this. It would be good to explore using them, and update this page!

Virtual machine

Install Visual Studio on a VM, and make sure the project builds and runs.

Friends

Ask a friend to check if it builds on their Windows machine!

CI/CD

It is possible to use CI/CD systems to check that a project builds. If it's not a graphical app, you can even test running it in the VM! An example setup for building .sln projects on push with GitHub Actions can be found here.

Do note that, while GitHub offers 2,000 minutes for free, using a Windows VM **counts as double minutes**! You only have 1,000 real minutes per month, which should still be largely enough.