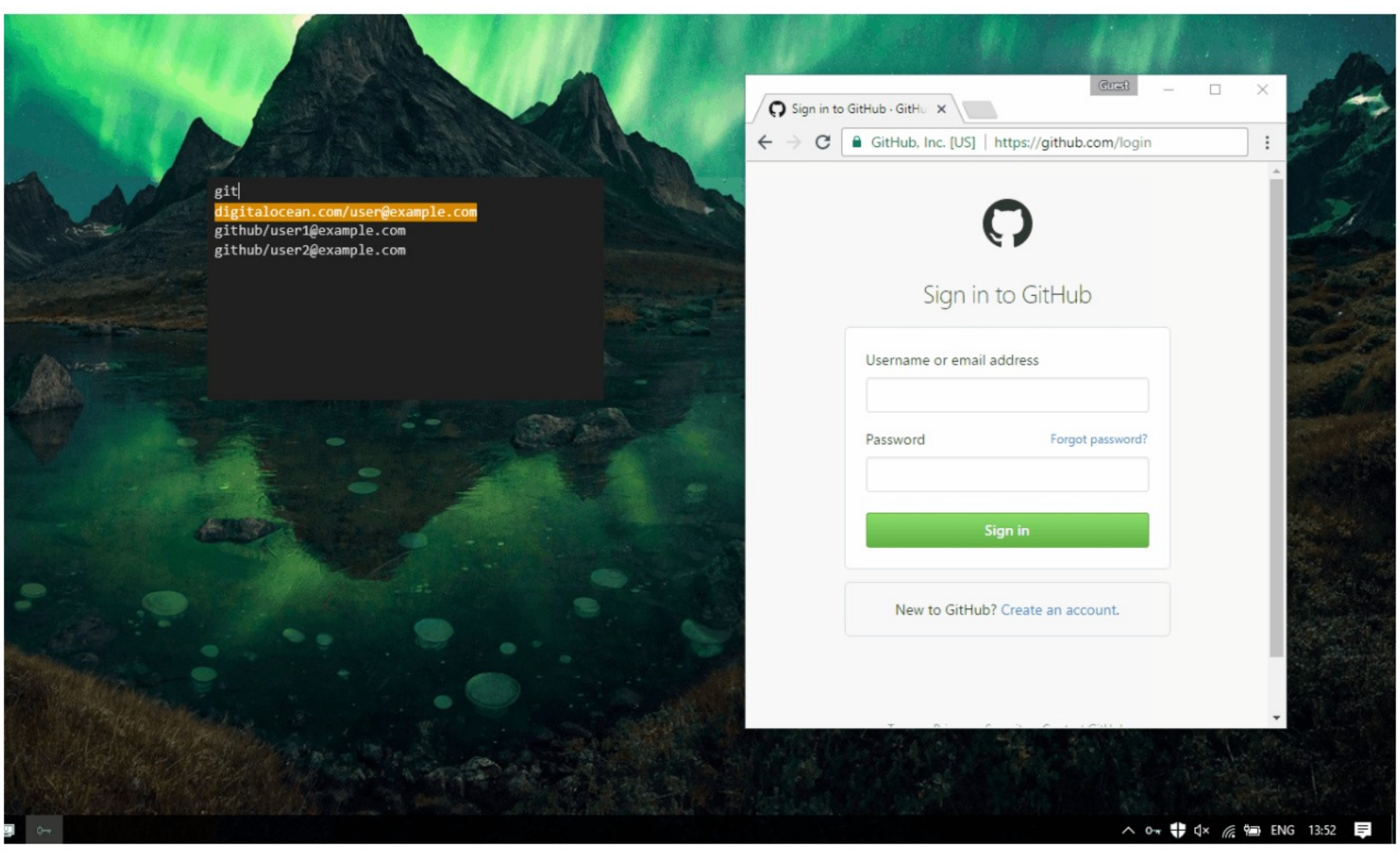
pass-winmenu

A simple, easy-to-use password manager for Windows.

Pass-winmenu follows the philosophy of (and is compatible with) the Linux password manager pass, which defines an open standard for password management that's easy to extend and customise to your personal requirements.



```
git
digitalocean.com/user@example.com
github/user1@example.com
github/user2@example.com
```

Sign in to GitHub · GitHu
GitHub, Inc. [US] | https://github.com/login
Sign in to GitHub
Username or email address
Password
Forgot password?
Sign in
New to GitHub? Create an account.
Guest
ENG
13:52

Donations to this project will go to acquiring a code signing certificate; for verifying downloads and allowing Windows Defender to eventually start trusting the application.

Donate

Introduction

Pass (https://www.passwordstore.org) stores passwords as GPG-encrypted files organised into a directory structure. Its simplicity and modularity offer many advantages, most importantly:

- Cryptography is handled by GPG, an open-source, high quality security suite trusted and used by security analysts, journalists, Linux distributions and many other parties all over the world
- The use of open standards makes it easy for anyone to develop compatible password managers for any platform they like (Linux, Android, Windows, Mac OS, etc)
- Because the passwords are simply stored as encrypted files in directories, you can organise them with your file manager and synchronise them across your devices using whatever method you prefer (Git, Dropbox, Nextcloud, etc).
- The passwords are encrypted with your GPG key, which can only be unlocked with your master password. A potential attacker will need to have your encrypted passwords, you GPG keys, *and* your master password to be able to do anything at all.

While many Linux integrations for pass are available, there are fewer options for Windows. Pass-winmenu aims to fill that gap. It allows for easy, keyboard-friendly interaction and has a minimal interface that stays out of your way.

Usage

Bring up the password menu with the keyboard shortcut `Ctrl Alt P`. The password menu allows you to quickly search through your passwords to find the one you are looking for. Navigate through the results by pressing Tab, and press Enter to decrypt the selected password.

The password will be decrypted using GPG, and your GPG key passphrase may be requested through pinentry. The decrypted password will then be copied to your clipboard and/or entered into the active window, depending on your `pass-winmenu.yaml` settings.

Configuration

Pass-winmenu can be configured using the `pass-winmenu.yaml` configuration file located next to the `pass-winmenu.exe` executable.

The configuration file is extensively documented, and there are many settings that can be changed to tweak the application to your liking, so take your time to look through it (you can find an example here). You can always generate a new configuration file, containing all settings and their default values, by renaming or deleting the old one and starting pass-winmenu.

Dependencies

Pass-winmenu is built against .NET Framework 4.6.2, which is included by default in Windows 10, and usually already installed on older Windows versions.

Git support is provided by LibGit2Sharp, which requires some native dependencies which are contained within the release builds.

For convenience, the release builds also contain a portable GPG installation, which pass-winmenu uses by default. If you already have GPG installed, you may want to use that instead. In that case, you can download the `nogpg` release, which will use your native GPG installation.

Installation

Installing pass-winmenu is as easy as downloading the zip file for the latest release and extracting it anywhere you want. It is recommended that you download the regular release, unless you already have GPG installed and accessible from your commandline. In that case you can also use the `nogpg` release. A Chocolatey package is also available.

If this is your first time using `pass`, you'll want to create a password store and import/create your GPG keys next. This process is explained below.

Setting up GPG:

If you already have a GPG key, you can skip this step and go to 'creating a new password store'. If you've never used GPG before, you can generate a new key. Start pass-winmenu, right click the key icon in the notification area, and click `Open shell`.

This will open a PowerShell window in which you'll be able to set up your GPG keys. Start by generating a new key:

```
powershell> gpg --gen-key
```

Follow the instructions to generate your GPG keys. You'll be asked to enter a passphrase, this is the passphrase that you will use to decrypt your passwords, so make sure it is secure enough.

Creating a new password store:

Determine in which directory you want to store your passwords. By default, pass-winmenu will assume it's `%USERPROFILE%\.password-store`. If you want to use that directory, create it:

```
powershell> mkdir $HOME\.password-store
```

Save the email address you used for creating your GPG key into a `.gpg-id` file in the root of your password directory. If you have multiple keys with the same email address, you can also use the key ID instead.

```
powershell> echo "myemail@example.com" | Out-File -Encoding utf8 $HOME\.password-store\.gpg-id
```

If you've used a different location for your password store directory, you'll have to point pass-winmenu to it. Open `pass-winmenu.yaml` in the directory where you've installed the application, and set the `password-store` variable to the correct location. Exit pass-winmenu if it was running, and start it again.

You should now have a working password manager.

Password synchronisation

If you want to access your passwords on multiple devices, you have several options. What follows are the instructions for setting up Git (which is by far the most popular option), but you can also use SVN, Dropbox, Google Drive, ownCloud, network shares, bittorrent sync, or anything else that synchronises files or provides access to them from multiple locations.

To synchronise your passwords using Git, initialise a new Git repository at the root of your password store:

```
powershell> cd $HOME\.password-store
powershell> git init
powershell> git add -A
powershell> git commit -m "Initialise password repository"
```

You'll also need a remote Git server. GitLab offers free private repositories, and GitHub does too for private accounts and up to three collaborators. Alternatively, you can of course run your own Git server.

Add an empty repository on your Git provider of choice, then connect your password store to it. It will usually come down to something like this:

```
powershell> git remote add origin https://github.com/yourusername/password-store.git
powershell> git push --set-upstream origin master
```

Accessing an existing password store on a different host

If you already have a password store and you want to access it from another computer, you'll have to import your GPG keys on it. Install pass-winmenu on your target PC, then export your GPG keys on the machine where you already have a working password store:

```
powershell> gpg --export-secret-key -a youremailaddress@example.com > private.key
```

Copy the `private.key` file to the machine on which you're setting up your password store, and import it.

```
powershell> gpg --import private.key
```

Now, set the key validity so that it can be used to decrypt your password files.

```
powershell> gpg --edit-key youremailaddress@example.com
gpg> trust
```

Set the trust level to `5` (ultimate trust) and save your key.

```
gpg> save
```

Clone your password repository

```
powershell> git clone https://github.com/yourusername/password-store.git $HOME/.password-store
```

Then run pass-winmenu, edit the generated `pass-winmenu.yaml` configuration file as necessary, and start it again.

Cross-platform support

Check out https://www.passwordstore.org/#other if you're looking for implementations for other operating systems.

Personally, I use Android Password Store for Android, and a dmenu script for Linux, which I've adapted from this script.