Canonical: Ubuntu Core Engineering Manager Take Home Test

Marc Magrans de Abril

Repository Location

The test results of the Ubuntu Core Engineering Manager Take Home Test are located in the `marcmagransdeabril/canonical-ubuntu-core-engineering-manager` repository.

Exercise 1

The solution can be found at exercise1 folder.

Assumptions

The Bash script makes several assumptions:

- Run the bash script on Ubuntu 22-04 LTS (minimal)
- The script stops after the first error
- The script logs the steps (-x)
- Builds the latest stable kernel version, but an existing image could improve the speed of the script
- Uses busybox to simplify the creation of the initrd image

Exercise 2

Instructions

The solution can be found at exercise2 folder.

The Shred function can be found shred.go and a some tests at shred_test.go

You can execute the tests:

```
cd exercise2
go test
```

Assumptions

The Shred function makes several assumptions about the file that it receives as input: * The file is a regular file. Otherwise the function returns an error. Note that for symbolic links this behaviour is debatable. * The file is readable and writable. Otherwise the function returns an error. * There is enough memory and free file descriptors to execute the function. * The file system supports the sync system call which flushes file data to disk and ensures that the data is written to persistent storage. * The file is not being written by another process. Otherwise the behaviour is platform specific, in general undefined. * Shred assumers that the file is not locked (if the OS allows it).

The file is not locked: Shred assumes that the file is not locked by another process, and can be overwritten.

Possible test cases

The possible test cases can be envisioned:

- (implemented) If a file does not exist, the Shred function should return an error.
- (implemented) If a file exists, after the execution of the function the file should not not longer be present.
- (implemented) If a big file exists (1 MB), after the execution of the function the file should not not longer be present.
- (implemented) Test that the function returns an error when the file is not a regular file (directory or pipe).
- (implemented) Test that the function handles file permissions correctly. For example, try running the function on a read-only file and ensure that it returns an appropriate error. Not all acess rights combinations tested.
- Test that the file is not recoverable after being shredded. You could use a data recovery tool to attempt to recover the shredded file and ensure that the recovered data is random and not the original file contents.