procfs

This package provides functions to retrieve system, kernel, and process metrics from the pseudo-filesystems /proc and /sys.

WARNING: This package is a work in progress. Its API may still break in backwards-incompatible ways without warnings. Use it at your own risk.

Usage

The procfs library is organized by packages based on whether the gathered data is coming from /proc, /sys, or both. Each package contains an `FS` type which represents the path to either /proc, /sys, or both. For example, cpu statistics are gathered from `/proc/stat` and are available via the root procfs package. First, the proc filesystem mount point is initialized, and then the stat information is read.

```
fs, err := procfs.NewFS("/proc")
stats, err := fs.Stat()
```

Some sub-packages such as `blockdevice`, require access to both the proc and sys filesystems.

```
    fs, err := blockdevice.NewFS("/proc", "/sys")
    stats, err := fs.ProcDiskstats()
```

Package Organization

The packages in this project are organized according to (1) whether the data comes from the `/proc` or `/sys` filesystem and (2) the type of information being retrieved. For example, most process information can be gathered from the functions in the root `procfs` package. Information about block devices such as disk drives is available in the `blockdevices` sub-package.

Building and Testing

The procfs library is intended to be built as part of another application, so there are no distributable binaries. However, most of the API includes unit tests which can be run with `make test`.

Updating Test Fixtures

The procfs library includes a set of test fixtures which include many example files from the `/proc` and `/sys` filesystems. These fixtures are included as a ttar file which is extracted automatically during testing. To add/update the test fixtures, first ensure the `fixtures` directory is up to date by removing the existing directory and then extracting the ttar file using `make fixtures/.unpacked` or just `make test`.

```
rm -rf fixtures
make test
```

Next, make the required changes to the extracted files in the `fixtures` directory. When the changes are complete, run `make update_fixtures` to create a new `fixtures.ttar` file based on the updated `fixtures` directory. And finally, verify the changes using `git diff fixtures.ttar`.