systemstat

Documentation online

systemstat is a package written in Go generated automatically by `gobi`.

systemstat allows you to add system statistics to your go program; it currently polls the linux kernel for CPU usage, free/used memory and swap sizes, and uptime for your go process, as well as the system you're running it on, and the system load. It can be used to make a crippled version of top that monitors the current go process and ignores other processes and the number of users with ttys. See the examples directory for go-top.go, which is my attempt at a top clone. Bear in mind that the intention of **systemstat** is to allow your process to monitor itself and it's environment, not to replace top.

Install (with GOPATH set on your machine)

- Step 1: Get the `systemstat` package

```
go get bitbucket.org/bertimus9/systemstat
```

- Step 2 (Optional): Run tests

```
$ go test -v bitbucket.org/bertimus9/systemstat
```

- Step 3 (Optional): Run example

```
$ cd to the first directory in your $GOPATH
$ cd src/bitbucket.org/bertimus9/systemstat
$ go run examples/go-top.go
```

##Usage

```
package main

import (
    "bitbucket.org/bertimus9/systemstat"
    "fmt"
)

var sample systemstat.MemSample

// This example shows how easy it is to get memory information
func main() {
    sample = systemstat.GetMemSample()
    fmt.Println("Total available RAM in kb:", sample.MemTotal, "k total")
    fmt.Println("Used RAM in kb:", sample.MemUsed, "k used")
    fmt.Println("Free RAM in kb:", sample.MemFree, "k free")
    fmt.Printf("The output is similar to, but somewhat different than:\n\ttop -n1 | grep Mem
}
```

##License