Nome: André Martins Ferreira
Cartão: 159098

Instalei a linguagem em um sistema Ubuntu 9.10.

1. Adicionei no arquivo ~/.bashrc as linhas:

export GOROOT=$HOME/go
export GOARCH=386
export GOOS=linux
export GOBIN=$HOME/bin

2. Rodei o comando sudo apt-get install bison gcc libc6-dev ed gawk make
3. sudo apt-get install mercurial
4. hg clone -r release https://go.googlecode.com/hg/ $GOROOT
5. cd $GOROOT/src
6. ./all.bash

Para compilar um programa:

1. 8g produtor1.go
2. 8l produtor1.8
3. ./8.out

Para se fazer uma computação em paralelo, é utilizado o comando go <expressão>, onde <expressão> é uma chamada de função. O comando go roda a expressão passada em uma nova “goroutine”, que é uma thread m-n leve.
A sincronização é feita através de channels, que são parecidos com pipes de unix, porém tipados. Durante a sua criação, além do tipo dos objetos que serão passados, pode ser definido o tamanho do buffer do canal, com 0 sendo o default. Quando um canal está cheio, qualquer rotina que tentar enviar algo por ele ficara bloqueada até uma outra rotina consumir um objeto do canal. Ao tentar receber, caso o buffer esteja vazio, a rotina bloqueia até que um objeto seja enviado ao canal.
ch := make(chan string, 10)
Criara um canal de strings bufferizado, com buffer de tamanho 10. A sua utilização é feita pelo operador <-, que possui uma versão binaria para o envio, e unaria para o recebimento.
ch <- "ola"
Envia para o canal ch o string "ola".
s := <- ch
Atribui a s o valor recebido pelo canal ch.