

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 produtorconsumidor.go`
2. `8l produtorconsumidor.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 thread, também chamada de "goroutine". Goroutines são multiplexadas dinamicamente pelo runtime em threads do sistema operacional, em um modelo m-n, com sua criação sendo mais leve do que a criação de uma pthread por exemplo.

A sincronização é feita principalmente 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 ficará 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)
```

Criar um canal de strings bufferizado, com buffer de tamanho 10. A sua utilização é feita pelo operador <-, que possui uma versão binária para o envio, e unária para o recebimento.  
`ch <- "ola"`

Envia para o canal `ch` o string "ola".

```
s := <- ch
```

Atribui a `s` o valor recebido pelo canal `ch`.

Anexo a esse relatório está um programa implementando produtores e consumidores através de um channel. Para compilar é provido junto um Makefile, além do programa já previamente compilado.

O programa deve ser usado com os seguintes argumentos: `./8.out -c <numero de threads consumidoras> -p <numero de threads produtoras> -b <tamanho do buffer>`. Onde `c` e `p` devem ser maiores do que 0, e `b` maior ou igual a 0.