

Tutorial de instalação de ferramentas

MC658 - PROJETO E ANÁLISE DE ALGORITMOS III

Docente: Prof. Cid Carvalho de Souza – PED: Natanael Ramos

INSTITUTO DE COMPUTAÇÃO – UNICAMP – 1º SEMESTRE DE 2019

1 Introdução

Esse documento é um tutorial para instalação das ferramentas que serão utilizadas na disciplina. Supõe-se que o sistema no qual serão feitas as instalações corresponde a alguma distribuição de GNU/Linux. Todos os comandos foram providos supondo que você esteja usando um terminal (ou emulador) para efetuar as instalações. Além disso, os comandos iniciam com um caractere \$ que não deve ser literalmente digitado, sendo este apenas um indicativo de que tal comando está sendo executado pelo terminal. Para cada aplicação é indicada a sua versão no momento em que o tutorial foi escrito. Não é mandatório que você instale a mesma versão, mas tenha em mente que os exemplos em sala e a correção dos trabalhos seguirão as versões aqui mencionadas.

1.1 Minizinc

Primeiro, efetue o download do Minizinc na página <https://www.minizinc.org/software.html>, na seção **Bundled binary packages (recommended)** baixe o arquivo correspondente ao item **64 bit Linux binary archive (Ubuntu 16 or later)**. No momento que este tutorial está sendo escrito, o Minizinc está em sua versão **2.2.3**, note que talvez seja uma nova versão quando você estiver fazendo essa instalação. Em um terminal, navegue até a pasta onde o arquivo `MiniZincIDE-2.2.3-bundle-linux-x86_64.tgz` foi baixado. Esse arquivo está compactado com extensão `tgz` e para descompactá-lo, execute o comando abaixo.

```
$ tar xf MiniZincIDE-2.2.3-bundle-linux-x86_64.tgz
```

Feito isso, será criada uma pasta de nome `MiniZincIDE-2.2.3-bundle-linux`. Para fins de simplicidade, nessa seção iremos nos referir a essa pasta como `$DIR`. Caso você prefira usar a IDE `MiniZincIDE` basta executar o script `MinizincIDE.sh` localizado dentro de `$DIR`. Caso contrário, para acessar o compilador `minizinc`, é necessário definir algumas variáveis de sistema para que as bibliotecas do Minizinc sejam encontradas pelo *linker*¹. Isso pode ser feito pela sequência de comandos apresentada a seguir (lembre-se de substituir `$DIR` pelo diretório onde o Minizinc foi descompactado).

¹<https://courses.cs.washington.edu/courses/cse378/97au/help/compilation.html>

```
$ export LD_LIBRARY_PATH=$DIR/lib:${LD_LIBRARY_PATH}
$ export QT_PLUGIN_PATH=$DIR/plugins
```

Isso deve ser feito em toda nova sessão antes de usar o `minizinc`. Caso deseje que tais variáveis fiquem permanentemente definidas em toda nova sessão, adicione as mesmas duas linhas no seu arquivo `.profile` (ou `.bash_profile`) localizado na sua pasta `home`. Para mais informações sobre variáveis de sistema, veja: <https://help.ubuntu.com/community/EnvironmentVariables>.

Agora, para executar o `minizinc`, basta invocá-lo da seguinte forma: `$DIR/bin/minizinc`. Caso queira digitar somente `minizinc`, adicione o caminho `$DIR/bin` à sua variável de sistema `PATH`, como mostrado abaixo. Essa variável contém os caminhos os quais o interpretador de comandos irá procurar pelos executáveis. Como mencionado previamente, caso queira que tal modificação seja permanente, adicione a mesma linha no seu arquivo `.profile`.

```
$ export PATH=$DIR/bin:${PATH}
```

Para testar sua instalação, você pode usar o modelo descrito a seguir. Salvando o modelo como `model.mzn`, basta executar o comando `minizinc model.mzn` para resolvê-lo.

```
var 1..3: x;
var 1..3: y;
constraint x+y > 3;
solve satisfy;
```

1.1.1 Documentação e tutoriais

Seguem links para alguns tutoriais, documentação oficial e outros recursos para aprender sobre o Minizinc:

- Tutorial oficial: www.minizinc.org/doc-2.2.3/en/part_2_tutorial.html
- *Handbook*: www.minizinc.org/doc-2.2.3/en/index.html
- Manual do usuário: www.minizinc.org/doc-2.2.3/en/part_3_user_manual.html
- Curso sobre Minizinc na plataforma *coursera*: <https://www.coursera.org/lecture/basic-modeling/1-1-1-first-steps-PYg0S>

1.2 Gurobi

O Gurobi oferece acesso livre para estudantes e para obtê-lo, basta criar uma conta no site www.gurobi.com e solicitar a licença em <https://user.gurobi.com/download/licenses/free-academic>. Feito isso, baixe o Gurobi **64-bit Linux** em www.gurobi.com/downloads/gurobi-optimizer. No momento em que este tutorial está sendo escrito, o Gurobi está em sua versão **8.10**. O arquivo baixado vem compactado em extensão `tar.gz`, então, utilize o mesmo procedimento de descompactação descrito para o Minizinc na Seção 1.1. Em seguida, na pasta descompactada, procure pelo executável `grbgetkey` no subdiretório `bin` e execute **dentro da**

rede da Unicamp o comando descrito abaixo, substituindo <KEY> pela chave de licença que você obteve no site do Gurobi.

```
$ grbgetkey <KEY>
```

Sendo \$DIR o diretório onde o Gurobi foi instalado, as variáveis de sistema descritas abaixo precisam ser definidas:

```
$ export GUROBI_HOME=$DIR
$ export PATH=${PATH}:${GUROBI_HOME}/bin
$ export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${GUROBI_HOME}/lib
```

Quando for definir a variável GUROBI_HOME, certifica-se de não utilizar o alias ~ para indicar sua pasta *home* e sim o caminho completo /home/<seu-nome-de-usuario>, uma vez que alguns pacotes que se baseiam nessa variável para buscar as bibliotecas do Gurobi não fazem a expansão do caractere ~. Para testar sua instalação, tente resolver o seguinte modelo:

```
$ gurobi_cl $DIR/linux64/examples/data/coins.lp
```

1.2.1 Documentação e tutoriais

Seguem links para alguns tutoriais, documentação oficial e outros recursos para aprender sobre o Gurobi:

- Exemplos: www.gurobi.com/resources/examples/example-models-overview
- Manual *quick start*: www.gurobi.com/documentation/8.1/quickstart_linux.pdf

1.3 Julia

Faça o download dos arquivos da instalação do Julia em: <https://julialang.org/downloads/> na seção **Generic Linux Binaries for x86** para **64-bit**. No momento em que este tutorial está sendo escrito, o Julia está em sua versão **1.1.0**. O arquivo baixado está compactado com extensão **tar.gz**, então siga as mesmas instruções de descompactação da Seção 1.1. Feito isso, denotaremos por \$DIR o diretório onde os arquivos foram descompactados. Para acessar os binários da instalação de qualquer local, adicione o caminho descrito no código abaixo na sua variável de sistema \$PATH.

```
$ export PATH=$DIR/bin:${PATH}
```

Em seguida, inicie o interpretador da Julia em um terminal, digitando **julia**. Os comandos a serem digitados no interpretador serão indicados por um caractere > no começo da linha. Iremos agora instalar o pacote JuMP.jl², o qual provê ferramentas para modelagem matemática utilizando Julia. Iremos usar a versão **0.18.5** do JuMP, uma vez que houve uma mudança

²<https://github.com/JuliaOpt/JuMP.jl>

significativa (principalmente na sintaxe) entre as versões 0.18 e 0.19³. A instalação pode ser feita da seguinte forma:

```
> import Pkg
> Pkg.add(Pkg.PackageSpec(name="JuMP", version="0.18.5"))
```

Em seguida, iremos instalar o pacote `Gurobi.jl`, o qual provê uma interface para as bibliotecas do Gurobi acessíveis através do JuMP. No momento em que este tutorial está sendo escrito, esse pacote está em sua versão **0.5.9**. Nota: **Para que a instalação funcione, é necessário que a variável de sistema `$GUROBI_HOME` esteja definida (Seção 1.2)**

```
> Pkg.add(Pkg.PackageSpec(name="Gurobi", version="0.5.9"))
> Pkg.build("Gurobi")
```

Para testar a instalação, execute a série de comandos que descrevem um modelo de programação linear apresentados a seguir:

```
> using JuMP, Gurobi
> model = Model(solver=GurobiSolver())
> @variable(model, 0 <= x <= 2)
> @variable(model, 0 <= y <= 30)
> @objective(model, Max, 5x + 3 * y)
> @constraint(model, con, 1x + 5y <= 3)
> solve(model)
```

A saída deve ser similar à seguinte:

```
Academic license - for non-commercial use only
Optimize a model with 1 rows, 2 columns and 2 nonzeros
Coefficient statistics:
  Matrix range    [1e+00, 5e+00]
  Objective range [3e+00, 5e+00]
  Bounds range    [2e+00, 3e+01]
  RHS range       [3e+00, 3e+00]
Presolve removed 1 rows and 2 columns
Presolve time: 0.00s
Presolve: All rows and columns removed
Iteration   Objective          Primal Inf.    Dual Inf.      Time
           0    1.0600000e+01   0.000000e+00   0.000000e+00    0s

Solved in 0 iterations and 0.00 seconds
Optimal objective  1.060000000e+01
```

1.3.1 Documentação e tutoriais

Seguem links para alguns tutoriais, documentação oficial e outros recursos para aprender sobre Julia e o pacote JuMP:

³<https://github.com/JuliaOpt/JuMP.jl/blob/master/NEWS.md>

- Página no próprio site da Julia com uma lista de recursos para aprender sobre a linguagem: <https://julialang.org/learning/>
- Documentação do JuMP: www.juliaopt.org/JuMP.jl/v0.18/
- Vídeo de um *workshop* mostrando uma visão geral de como usar Julia para resolver problemas de otimização: www.youtube.com/watch?v=nnL7yLMVu6c

1.4 Gnumeric

Para instalar o Gnumeric, utilize o gerenciador da sua própria distribuição Linux. Se você está usando o Ubuntu, por exemplo, execute:

```
$ sudo apt install gnumeric
```

Caso você não tenha permissões de `sudo`, você pode tentar compilar o código fonte do Gnumeric: <http://ftp.gnome.org/pub/GNOME/sources/gnumeric/1.12/gnumeric-1.12.28.tar.xz>. As instruções de compilação e instalação estão no próprio arquivo. A versão utilizada na correção dos trabalhos será a **1.12.28**.

1.4.1 Documentação e tutoriais

- Guia oficial: <https://help.gnome.org/users/gnumeric/stable/gnumeric.html> (A Seção 7. *The Solver* é a mais importante)