

Laboratórios de Informática I

2020/2021

Mestrado Integrado em Engenharia Informática



Universidade do Minho

Sessão Laboratorial 1

Programa

Utilização de ferramentas de desenvolvimento de *software*:

- sistemas operativos baseados em UNIX e linha comandos
- interpretador/compilador de Haskell
- sistema de controle de versões
- documentação do código (Haddock)
- teste de *software*
- ...

Realização de projecto de programação (Haskell)

- projectos anos anteriores:
<https://haslab.github.io/Teaching/LI1/>

Funcionamento

- Aulas laboratoriais (com guiões semanais)
- Recomenda-se que tragam portátil pessoal para as aulas
- Horário de apoio:
 - Daniel Maia: Segundas, 15h-16h
 - Francisco Ribeiro: Quintas, 17h-18h
 - Jácome Cunha: Terças, 14h-15h; Quartas, 15h-16h
 - Rui Rua: Sextas, 15h-16h

Avaliação

Nota final = Nota do projecto [1-20] × Prestação individual [0-1]

- projecto composto por duas fases
- prestação individual:
 - presença e empenho nas aulas
 - contributo para a realização do projecto (por auto-avaliação)
 - prestação na defesa.
- **mais de 4** faltas implica prestação individual = 0
- **fraude** implica prestação individual = 0 e comunicação à direcção de curso

Acerca do Projecto

- realizado em 2 fases em **grupos de 2**
- enunciado da primeira fase é apresentado na 3ª semana de aulas
- prazo de entrega da primeira fase é **5 de dezembro às 23:59h (Portugal continental)**
- enunciado da segunda fase é apresentado na 9ª semana de aulas
- prazo de entrega da segunda fase é **23 janeiro às 23:59h (Portugal continental)**
- defesas serão na semana de 25 a 30 de Janeiro

Sistema de Ficheiros em UNIX

Ficheiros organizados numa hierarquia

- informação é armazenada sob a forma de *ficheiros* e *diretorias*
- directorias formam uma árvore a partir da *root* – representada por `/`
- ao utilizador é atribuída uma parte dessa árvore, a sua *home* – e.g. `/home/HaskellCurry`
- ficheiros podem ser de *texto* puro, *aplicações*, ou *dados de aplicações* – e.g. `/home/HaskellCurry/LI2017/aula1.pdf`
- ficheiros são distinguidos pela sua *extensão* – e.g. `ficha1.hs`, `relatorio.tex`, `relatorio.pdf`

Linha de Comandos (CLI)

É uma forma de interação com o sistema operativo, permitindo, e.g.

- navegar no sistema de ficheiros
- compilar e executar programas

Porque precisamos da linha de comandos?

- a Graphical User Interface, ou simplesmente GUI, oculta detalhes do programa e pode não suportar todas as funcionalidades
- interface gráfica é muitas vezes apenas uma “capa” para uma aplicação que corre na linha de comandos
- aceder a um servidor que nem sequer tem interface gráfica
- com alguma prática, a execução de algumas tarefas é mais rápida a partir da linha de comandos

Comandos básicos para interagir com o sistema de ficheiros

Navegar

`pwd` mostra o caminho para directoria actual

`ls` lista o conteúdo da directoria actual

`cd` muda de directoria

Criar, modificar e remover

`mkdir` cria uma directoria

`cp` copia ficheiros

`mv` move ficheiros

`rm` remove ficheiros (**permanente!!**)

`rmdir` remove directoria vazia

Código Fonte

Aplicações são criadas através da compilação de ficheiros *código fonte*

- ficheiros de texto puro – e.g. `ficha1.hs`
- apenas caracteres são armazenados (alguns não impressos, como mudanças de linha e tabulações)
- contrastam com documentos de processadores de texto – e.g. `relatorio.doc`

Editores de Texto

O que são?

- peças fundamentais no desenvolvimento de aplicações
- permitem criar e modificar qualquer tipo de código fonte (ou qualquer outro tipo de ficheiro de texto)

Tipo de editores

- editores mais básicos apenas apresentam conteúdo do ficheiro – e.g. *Notepad* – pouco recomendados
- editores mais avançados têm funcionalidades como *syntax-highlighting*, gestão de indentação, *auto-complete*, *bracket matching*, etc. – e.g. Emacs, Vim, Sublime

Atenção: editores de texto não são processadores de texto

Ambientes de Desenvolvimento

Conjunto de ferramentas para suportar o desenvolvimento de programas:

- compiladores, gestores de bibliotecas, geradores de documentação, etc.
- editores podem também estar integrados no processo (*Integrated Development Environment*, IDE)

Linguagem de Programação

Haskell

- linguagem funcional lecionada na disciplina de *Programação Funcional* (PF)
- a *Haskell Platform* (Windows) ou o *ghcup* (Linux, Mac OS) contêm as ferramentas necessárias
 - compilador (`ghc`)
 - interpretador (`ghci`)
 - gestor de bibliotecas (`cabal`)
 - gerador de documentação (`haddock`)
 - ...

Sessão Laboratorial 1

- Guião sobre o *Ambiente de Desenvolvimento*
- Disponível no site de elearning