

UNIVERSIDADE FEDERAL DE SANTA MARIA
Curso de Ciência da Computação e Sistemas de Informação
Disciplina de Paradigmas de Programação
Trabalho final de Haskell
Professora: Juliana Kaizer Vizzotto

0.1 Descrição

Neste trabalho você deve re-implementar os seguintes programas de utilidade do Unix em Haskell: `cat`, `sort`, e `uniq`, chamando-as de `hcat`, `hsort`, e `huniq`, respectivamente. Você não precisa implementar todas as opções desses programas, entretando cada um deles deve ler a entrada de um arquivo (ou arquivos, no caso de `hcat`) ou da entrada padrão, e todos devem imprimir na saída padrão. Utilize um algoritmo de quicksort para o programa `hsort`. Nenhum dos programas deve adicionar linhas extras nos arquivos originais.

Leia as páginas de manual do próprio Unix para `cat`, `sort`, e `uniq`. Basicamente, `cat` copia (concatena) arquivos no stream de saída, `sort` ordena as linhas em um arquivo, e `uniq` linhas duplicadas consecutivas em um arquivo (somente as consecutivas).

Para os programas `hsort` e `huniq`, defina as funções de ordenação e remoção de repetidas em funções separadas que não sejam do tipo IO. Devido à avaliação lazy do Haskell você pode ter um código eficiente aplicando essas funções em uma lista contendo todas as strings do arquivo ao invés de processá-las diretamente no código de IO. Este é um dos benefícios da avaliação lazy: a possibilidade de decompor programas de maneira eficiente.

0.2 Funções Uteis

- `putStr` e `putStrLn`
- `getChar`
- `getLine`
- `hsetBuffering`
- `openFile`
- `getContents`
- `hGetContents`
- `hClose`
- `System.getArgs`
- `mapM`
- `interact` (o quê `interact id` faz?)
- `lines`
- o operador `(!!)`, que permite o acesso aos índices em listas.