Gerador de índice remissivo de um texto

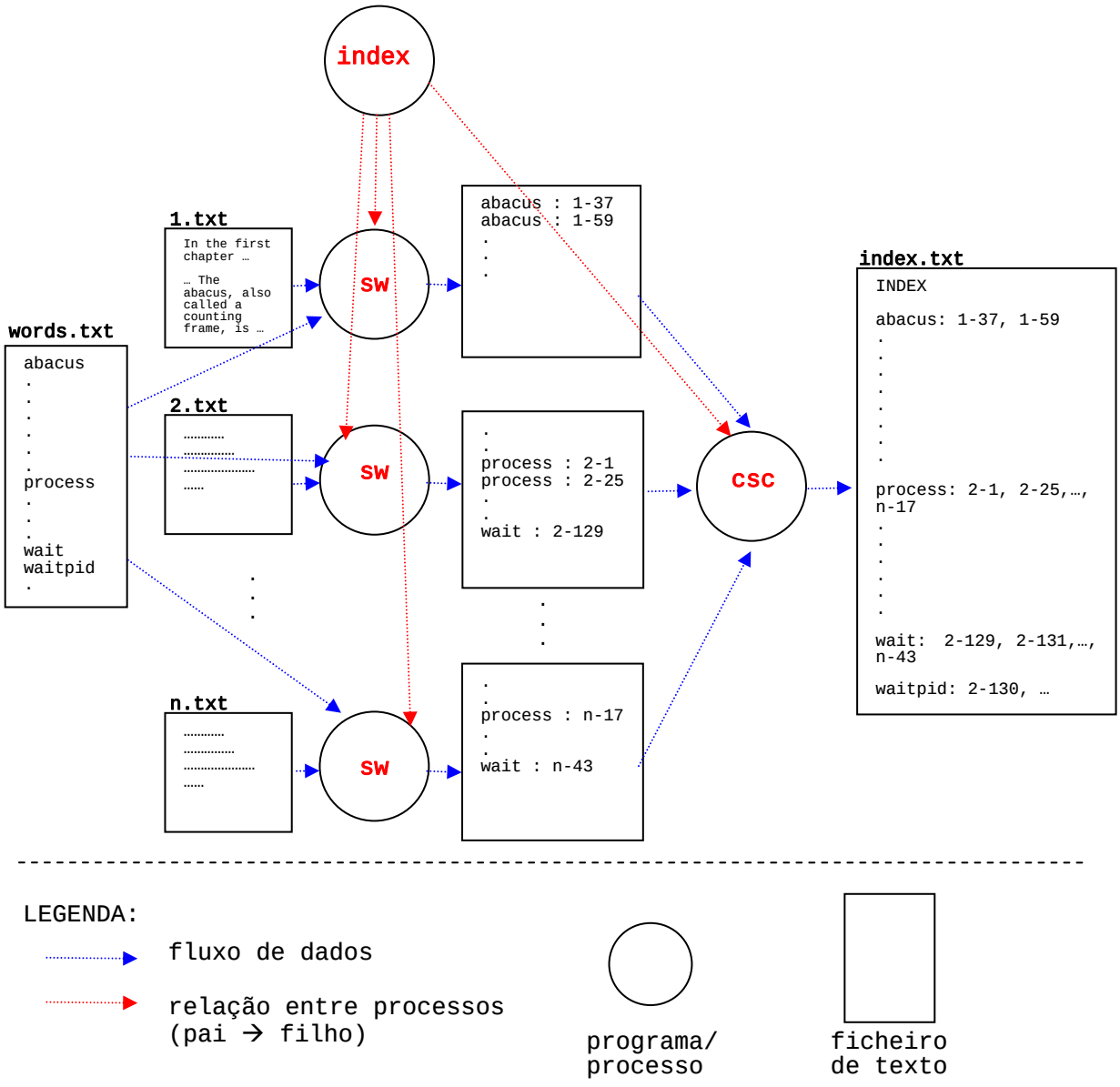
Objetivos

O objetivo final deste trabalho é desenvolver uma aplicação que permita construir o índice remissivo de um texto, a partir de um conjunto de ficheiros que compõem esse texto e de uma lista de palavras a indexar.

Através da sua realização, pretende-se proporcionar a familiarização com a programação de sistema, em ambiente Linux, envolvendo, principalmente, a manipulação de ficheiros e diretórios, o desenvolvimento de aplicações multiprocesso, a utilização de *pipes* como mecanismo de comunicação entre processos e invocação de programas externos.

Especificação do trabalho

- Apresenta-se a seguir um esquema da aplicação multiprocesso a desenvolver, bem como das suas entradas, saídas e resultados intermédios de processamento:



- O código executável da aplicação a desenvolver terá o nome **`index`** e será lançado em execução através do comando **`index <dir>`** em que **`<dir>`** representa o nome de um diretório que deverá conter todos os ficheiros a processar.
- Estes ficheiros, que constituem as entradas da aplicação, são:
  - O texto cujo índice remissivo se pretende construir, que deverá estar distribuído por diversos ficheiros de texto simples.
  - Uma lista das palavras a indexar, no índice remissivo. As palavras devem estar guardadas num ficheiro de nome **`words.txt`**, no formato "uma palavra por linha".
- A aplicação deve começar por verificar a existência, nesse diretório, dos ficheiros a processar, terminando imediatamente se não encontrar o ficheiro **`words.txt`** ou ~~se não~~ os ficheiros do texto a indexar.
- A saída da aplicação deve ser constituída por um ficheiro de texto, de nome **`index.txt`**, guardado no diretório indicado como parâmetro da aplicação, em que cada palavra é seguida da lista das suas ocorrências, no formato:

  `<palavra>: <ficheiro>-<linha>, <ficheiro>-<linha>, ...`

  - Nota: **`<ficheiro>`** não deve englobar a extensão do ficheiro que poderá ou não existir. Para simplificar o código, sugere-se que tratando-se, por exemplo, de um livro, os capítulos sejam gravados em ficheiros com os nomes **`1.txt`**, **`2.txt`**, **`3.txt`**, etc.
  - Ilustra-se a seguir o formato do ficheiro **`index.txt`** num caso desses (ver exemplo no Moodle):

```
abacus: 1-37, 1-59
...
wait: 2-129, 2-131, 3-25...
...
```

- O processo inicial (**`index`**) deve lançar em execução **`n`** processos **`sw`** (*search words*), um por cada ficheiro a processar. Cada um dos processos **`sw`** deve procurar no seu ficheiro de texto as palavras a indexar, e, por cada ocorrência, escrever num ficheiro de saída: a palavra, o nome do ficheiro de texto (sem extensão) e a linha onde a palavra foi encontrada, usando o formato:
  - **`<palavra> : <ficheiro>-<linha>`**
  - exemplo: se a palavra **`wait`** for encontrada no ficheiro **`2.txt`**, nas linhas 129 e 131, no ficheiro de saída do processo **`sw`** que processar **`2.txt`** deveria ser escrito, além de outro, o seguinte texto:

```
wait : 2-129
wait : 2-131
```

- Depois de todos os processos **`sw`** terem terminado, o processo **`index`** deve lançar em execução o programa **`csc`** (*concatenate, sort and clean*) que deve juntar todos os ficheiros resultantes dos processos **`sw`**, ordenar o resultado, tendo em conta a ordem alfabética das palavras indexadas, e "limpar" o texto resultante, agregando a informação relativa às ocorrências de cada palavra. O resultado deste processamento deve ser guardado no ficheiro **`index.txt`**, no formato anteriormente ilustrado.

Notas sobre o desenvolvimento

- Começar por desenvolver os programas **`sw`** e **`csc`**.
- O programa **`sw`** deve recorrer ao utilitário **`grep`** para pesquisar as palavras no seu ficheiro.
- O programa **`csc`** deve recorrer aos utiliários **`cat`** e **`sort`**, para concatenar os ficheiros e proceder à ordenação do ficheiro resultante.
- Fazer testes de erro nas chamadas ao sistema e usar sempre a opção de compilação **`-Wall`**, por forma a garantir que a compilação dos programas não dá origem a avisos (*warnings*).
- Não podem ser usadas chamadas "system", nem o equivalente execlp("sh","sh","-c", …).
- A formatação dos ficheiros intermédios (auxiliares) pode ser escolhida livremente.

Entrega do trabalho

- Data limite para a entrega do trabalho: 2015/04/19, às 23:55h.
- Oportunamente serão publicadas algumas regras para a entrega do trabalho, na página de "Sistemas Operativos", no Moodle da Universidade do Porto.