Sistemas Operativos

Processamento e análise de medições de temperatura

Universidade do Minho

16 de outubro de 2024

Informações gerais

- Cada grupo deve ser constituído por até 3 elementos;
- O trabalho deve ser entregue até às 23:59 de 10 de dezembro;
- Deve ser entregue o código fonte e um relatório de até 10 páginas de conteúdo (A4, 11pt) no formato PDF (não são contabilizadas capas e anexos para o limite de 10 páginas), justificando a solução, nomeadamente no que diz respeito à arquitetura de processos e à escolha e uso concreto dos mecanismos de comunicação;
- O trabalho deve ser realizado tendo por base o sistema operativo Linux como ambiente de desenvolvimento e de execução;
- O trabalho deve ser submetido num arquivo Zip com nome `grupo-xx.zip`, em que `xx` deve ser substituído pelo número do grupo de trabalho (*p.ex.*, `grupo-01.zip`) via Blackboard;
- A apresentação do trabalho ocorrerá em data a anunciar, previsivelmente entre os dias 12 e 13 de dezembro;
- O trabalho representa 50% da classificação final da unidade curricular de Sistemas Operativos. Tenha em conta que o trabalho será avaliado na sua globalidade mas a classificação resultante será sempre individual, dependendo dos contributos de cada na realização do projeto e na discussão que terá lugar na sessão de apresentação. As entregas com atraso serão penalizadas em 2 valores (em 20) por cada bloco de 24h.

Resumo

Pretende-se implementar uma solução para o processamento e análise de dados de sensores de temperatura localizados em diferentes regiões do país. Esta solução baseia-se em três programas a desenvolver:

- O programa *sort* ordenará os dados de uma dada região por ordem crescente num ficheiro binário que conterá os dados obtidos de todas as regiões.
- O programa *stats* calculará estatísticas, como a média, mediana, valor máximo e mínimo, sobre os dados de uma região, a partir do referido ficheiro binário.
- O programa *report* agregará e calculará estatísticas de várias regiões, indicando as regiões que se destacam em cada uma das estatísticas.

Ficheiro de dados

Neste projeto, considere um ficheiro de dados *binário* composto por duas partes. Um cabeçalho, com dois valores inteiros que indicam o número de regiões no ficheiro e o número de registos por região, por esta ordem. De seguida, os valores inteiros que correspondem às medições de todos os sensores de todas as regiões.

Todas as regiões têm o mesmo número de registos. Os primeiros N registos correspondem à região com identificador 1, os segundos N à região com identificador 2 e assim sucessivamente. No exemplo ilustrado na Figura 1 o número de registos por região é de 200 000 por isso, terá de ter em conta que esse valor poderá ascender mesmo a milhões de registos.

Nota: A equipa docente disponibilizou o código-fonte do gerador deste ficheiro. Este recebe como argumentos o número de regiões, o número de registos por região e o nome do ficheiro onde serão escritos os valores.

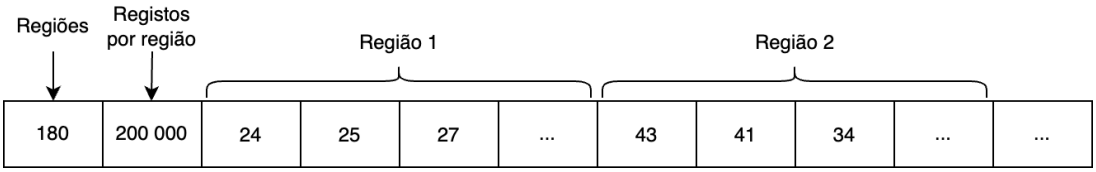

Figura 1: Exemplo de representação visual do ficheiro em binário.

```
$ ./generate_sensor_data <num_regions> <num_region_records> <output_file>
```

Pode e deve utilizar ficheiros com diferentes configurações para testar o seu projeto. Junto com o enunciado pode encontrar o ficheiro de exemplo *sensor_data.bin*, gerado por este programa que conta com 180 regiões, cada uma com 200 000 medições.

Programa *sort* (5 valores)

O programa *sort* recebe como primeiro argumento o caminho para o ficheiro de dados que contém todos os registos de todas as regiões. Como segundo argumento recebe o identificador de região cujos registos devem ser ordenados por ordem crescente.

```
$ ./sort <sensor_data_file> <region>
```

Deve assumir que a memória disponível não permite que os registos de uma dada região sejam carregados totalmente para memória. **Deve ter em conta que a ordenação dos registos da região deve ser realizada sem recurso à criação de ficheiros temporários.**

Programa *stats* (3 valores)

O programa *stats* recebe como primeiro argumento o caminho para o ficheiro de dados que contém todos os registos **ordenados** de todas as regiões. Como segundo argumento, recebe o identificador de região para a qual irá calcular as seguintes estatísticas: média, mediana, valor máximo medido, valor mínimo medido.

```
$ ./stats <sensor_data_file> <region>
```

Este programa deverá **tirar partido do programa *sort* para ordenar os registos de uma dada região** e, posteriormente, escrever as estatísticas num ficheiro binário, um por região. Assegure que o nome de cada ficheiro identifica a região à qual pertence, por exemplo, *region-002-stats.bin*. Cada ficheiro deve conter a seguinte estrutura de dados, que reúne estatísticas computadas para a região correspondente:

```
typedef struct region_stats {
    int region_id;
    int median;
    float average;
    int max;
    int min;
} region_stats;
```

Programa *report* (5 valores)

O programa *report* recebe como argumento o número de regiões existentes para gerar o relatório final. A computação de estatísticas das regiões em análise deve recorrer à **execução do programa** *stats*, para cada região. Paralelize a computação das estatísticas das regiões, sendo que o resultado de cada região deve ser comunicado via **pipe anónimo** ao processo principal[1].

```
$ ./report <num_regions>
```

[1] Assuma que o *output* do programa *stats* pode ser escrito concorrentemente num único *pipe anónimo*.

Este programa deverá imprimir para o *stdout*:

- A região com o valor mais alto, bem como o valor medido;
- A região com o valor mais baixo, bem como o valor medido;
- A região com o menor valor médio, bem como o valor medido;
- A região com o maior valor médio, bem como o valor medido;
- As estatísticas de cada região;

Veja o seguinte exemplo, para o caso de duas regiões:

```
A Região 1 registou o valor mais alto: 45.2 °C;
A Região 2 registou o valor mais baixo: 25.3 °C;
A Região 2 registou o valor médio máximo: 37.6 °C;
A Região 2 registou o valor médio mínimo: 26.6 °C;

Região 1:
    * Valor médio: 32.3 °C
    * Mediana: 34 °C
    * Valor máximo: 45.2 °C
    * Valor mínimo: 26 °C

Região 2:
    * Valor médio: 37.6 °C
    * Mediana: 37 °C
    * Valor máximo: 38 °C
    * Valor mínimo: 25.3 °C
```

Valorização (5 valores)

- Evite que o programa *stats* crie ficheiros temporários binários para serem consumidos pelo programa *report*. Para este efeito, adicione um novo argumento no programa *stats* para indicar que o output gerado deve ser escrito para o *stdout*.
- Permita que o utilizador limite o número de processos a computar estatísticas das regiões em simultâneo, no programa *report*, adicionando um novo parâmetro ao programa para este efeito — `max_processes`.

Estrutura de código (2 valores)

Terá de implementar o programa em C e de usar as chamadas ao sistema estudadas na unidade curricular. Deverá ainda estruturar o seu código de forma modular, usando header files (ficheiros cabeçalho), definindo e declarando todas as funções necessários e ainda automatizar o processo de compilação e de geração de programas recorrendo à definição de uma Makefile.