

Trabalho Prático de Sistemas Operativos

Serviço de Contagem

Grupo de Sistemas Distribuídos
Universidade do Minho

29 de Abril de 2014

Informações gerais

Este texto descreve o trabalho prático de Sistemas Operativos, LEI e LCC, a realizar por grupos de alunos não excluídos por excesso de faltas. É nesta altura que os alunos que obtiveram classificação positiva no ano lectivo anterior devem decidir se pretendem realizar novo trabalho, eliminando a classificação anterior, ou reutilizar a classificação do ano passado, eventualmente truncada a 15 valores. Em breve serão dadas instruções como se processa esta opção, assim como a forma de cada grupo indicar os alunos que o constituem.

Quem realizar o trabalho deverá ter em atenção que:

- Cada grupo deve ser constituído por até três elementos, sem mistura de alunos com e sem nota do ano anterior.
- O trabalho tem de ser entregue até 2014-06-02 às 23:59:59. Não deixe para o último minuto!
- Deve ser entregue um ficheiro zip com o código fonte, dados de teste e um relatório que explique as opções tomadas e os testes efectuados.
- A apresentação será no dia 2014-06-02 para LCC e nos dias 4 e 6 para LEI. Oportunamente serão disponibilizados no blackboard grupos de horários para apresentação dos trabalhos, devendo cada grupo inscrever-se num "slot" livre.

Resumo

O trabalho consiste na implementação, em linguagem C, de um *serviço de contagem*. Este deverá permitir registar persistentemente incrementos em contadores, bem como reportar valores de contadores, individuais ou agregados. Cada contador tem um nome hierárquico, constituído por uma sequência de strings; por exemplo, num cenário de identificação de casos de gripe, o nome poderia representar o triplo distrito-concelho-freguesia, e.g., ["Braga", "Braga", "S. Victor"]. Assuma que existem muitos contadores, que cada nome tem pelo menos dois componentes, e que existem poucos valores diferentes para o primeiro componente.

Descrição

O serviço deverá ser constituído por: 1) um programa **servidor**, que recebe pedidos de registo ou reporte e processa os pedidos, delegando para processos filho, de modo a aproveitar potencial concorrência; 2) uma biblioteca para ser usada por processos “clientes”, que comunicam com o servidor. O programa servidor deverá permitir:

- registar incrementos num determinado contador; e.g., acrescentar 100 casos de gripe à freguesia [Braga, Guimarães, Pevidém];
- reportar o total agregado até um determinado nível, de todos os contadores que tenham um determinado prefixo (e.g., total de casos no distrito de Braga, discriminados por freguesia). Assuma que é passada uma sequência de strings com pelo menos um componente.

Considere que o serviço deve registar os incrementos em suporte persistente (ficheiros), à medida que chegam, devendo no entanto conseguir responder aos pedidos de agregação de um modo eficiente, mantendo em memória a informação relevante.

O sistema deve estar preparado para reiniciar processos filhos, caso estes terminem inesperadamente (e.g., um crash), devendo continuar a responder correctamente a pedidos, tendo em conta todo o registo persistente.

A biblioteca deverá disponibilizar as seguintes funções:

```
int incrementar(char *nome[], unsigned valor);
int agregar(char *prefixo[], unsigned nivel, char *path);
```

Tanto **nome** como **prefixo** são arrays de strings terminados por um endereço nulo (à semelhança do **argv** recebido pela função **main**).

O argumento **path** corresponde a um ficheiro onde o resultado será escrito, devendo a função bloquear até tal ter ocorrido.

O parâmetro **nivel** indica a discriminação desejada a partir do prefixo especificado. Por exemplo, para o prefixo [Braga]: ao nível 0 corresponderá ao agregado de todo esse distrito; ao nível 1 corresponderá o conjunto de valores discriminados por concelho; e ao nível 2 o total por freguesia.

O formato dos ficheiros de resultados deverá ser orientado à linha, cada uma delas composta pelo nome completo do agregado seguido do seu valor, separados por ‘:’. Por exemplo, para o prefixo [Braga], nível 1:

```
Braga:Amares:100
Braga:Barcelos:123
Braga:Braga:205
Braga:Cabeceiras de Basto:76
Braga:Guimarães:21
...
```

no agregar SÓ RECEBE o distrito
o resto o nível é que define!

SIGCHLD -> pid=wait()