

# Projeto de SO - Stream Processing

## Grupo 96

Sérgio Jorge (A77730)      Vítor Castro (A77870)      Marcos Pereira (A79116)

### Resumo

Neste relatório faremos uma análise do projeto de Sistemas Operativos, no qual o objetivo era desenvolver um programa, em C, de *Stream Processing*. Assim, este documento apresenta detalhadamente a abordagem tomada ao problema proposto pela equipa docente da UC.

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Problema</b>	<b>2</b>
<b>3</b>	<b>Solução</b>	<b>2</b>
3.1	const . . . . .	2
3.2	filter . . . . .	2
3.3	window . . . . .	2
3.4	spawn . . . . .	3
3.5	controlador . . . . .	3
<b>4</b>	<b>Conclusão</b>	<b>4</b>

## 1 Introdução

Este projeto foi realizado com o objetivo de desenvolver um programa responsável pela conceção de um sistema de *Stream Processing*. Foram, então, propostas pelos professores, a realização de quatro componentes computacionais, construídas em C, que produzem determinadas operações elementares e a realização de um controlador que gere a rede de processamento. A realização das componentes assim como do próprio controlador, permitiram melhorar e consolidar os conhecimentos adquiridos, durante o semestre, na UC de Sistemas Operativos. Assim, de modo a facilitar a compreensão do projeto, o relatório está dividido da seguinte forma:

**Secção 2 :** Problema;

**Secção 3 :** Solução;

**Secção 4 :** Conclusão.

## 2 Problema

Neste projeto de SO, é-nos pedido para fazermos um sistema de *Stream Processing*. Assim, o que devemos fazer para gerar a rede de processamento são os seguintes módulos:

### 1 - **const** <valor >

Reproduz as linhas e acrescenta uma nova coluna com o valor que é passado como argumento à função;

### 2 - **filter** <coluna ><operador ><operando >

Reproduz as linhas nas quais a condição dada como argumento se verifica;

### 3 - **window** <coluna ><operação ><linhas >

Reproduz todas as linhas e acrescenta uma nova coluna com o resultado de uma operação sobre os valores da coluna indicada nas linhas anteriores;

### 4 - **spawn** <cmd ><args...>

Reproduz todas as linhas, executando o comando indicado uma vez para cada uma delas e acrescentando uma nova coluna com o valor do exit status;

### 5 - **controlador**

A função do controlador é a de gerir a criação e manutenção de nós e das ligações entre eles.

## 3 Solução

### 3.1 **const**

- guardamos o valor que é passado como argumento;
- definimos um array para o qual lemos o que é inserido no stdin;
- colocamos o carácter "dois pontos" no final do array;
- inserimos o valor no fim do array;
- escrevemos o array para o stdout.

### 3.2 **filter**

- guardamos o valor dos argumentos (operador e duas colunas);
- definimos um array para o qual lemos o que é inserido no stdin;
- guardamos os valores que estão nas duas colunas que nos foram indicadas nos argumentos da função;
- fazemos a comparação entre os valores guardados na alínea anterior;
- se a comparação for válida, escrevemos a linha para o stdout. Se não for válida, nada é feito.

### 3.3 **window**

- guardamos o valor dos argumentos (coluna, operador e linhas);
- definimos um array A com linhas posições e pomos esse array a zeros;
- definimos um array B para o qual lemos o que é inserido no stdin;

- guardamos o valor que está na coluna do stdin que nos foi indicada no argumento da função;
- preenchemos o array A com os valores que vão sendo lidos pela alínea anterior. Este array só tem linhas posições pelo que quando enche, é substituído o valor que está na primeira posição;
- para as diferentes operações, percorre-se o array A e guarda-se o resultado num array Output;
- colocamos o carater "dois pontos" no final do array B;
- inserimos o array Output no fim do array B;
- escrevemos o array B para o stdout.

### 3.4 spawn

- guardamos o valor dos argumentos que têm o carater "\$" na frente para o array A e a posição destes para o array P;
- definimos um array B para o qual lemos o que é inserido no stdin;
- percorremos o array A que contém os valores das colunas a serem lidas e copiamos o que está na coluna para a posição de argv onde o carater "\$" foi encontrado;
- faz-se um fork que executa através dos argumentos da função que foram entretanto manipulados;
- guarda-se o resultado da execução num array Output;
- colocamos o carater "dois pontos" no final do array B;
- inserimos o array Output no fim do array B;
- escrevemos o array B para o stdout.

### 3.5 controlador

O controlador gere a criação de nós e injects, e das ligações entre estes. Para esse efeito, são guardadas informações em 5 arrays globais:

- nodes - Guarda os process id's de todos os nós em execução
- connections - Guarda os process id's de todos os processos "ponte" em execução (explicados abaixo)
- connectionDests - Guarda, relativamente a cada nó, a lista de nós que estão a receber o seu output
- injects - Guarda os process id's de todos os injects em execução
- injectConnections - Guarda os process id's de todos os processos "ponte" em execução que ligam um inject a um nó

Por forma a realizar as ligações entre nós, foi necessária a criação de um processo auxiliar, denominado connect. Cada nó tem um destes processos ponte associado a si, que copia o seu output e envia para todos os nós a que está conectado.

Para tornar possíveis estas ligações foi necessária a criação de um par de named pipes (FIFOs) para cada nó, que substituem o seu stdin e stdout. Assim, um processo ponte consegue introduzir informação em um nó e obter o seu output através destas pipes. Um processo inject tem a si

associado apenas uma named pipe, que substitui o seu stdout (um inject nunca recebe dados de outro nó).

Quando uma das conexões entre nós é modificada, o processo ponte anterior (caso exista) é terminado com SIGTERM (de modo a não serem perdidos dados) e é feito waitpid() para aguardar que termine. De seguida, é usada a informação guardada em connectionDests para juntar os novos ouvintes aos anteriores e criar um novo processo ponte que escreva em todos.

## 4 Conclusão

Este projeto serviu para aprofundarmos o conhecimento da linguagem C, assim como as bibliotecas que lhe estão associadas. Achámos que, a realização de um trabalho deste tipo permite uma consolidação proveitosa da linguagem, e dos conceitos lecionados nas aulas de Sistemas Operativos. Permite também melhorar as habilidades na resolução de problemas. cremos ter sido vantajoso e engraçado ter feito este trabalho apesar de também sermos da opinião de que podíamos, apesar de tudo, ter tido uma melhor performance se tivéssemos mais tempo e menos carga de trabalhos por parte das outras UCs.