

# **Relatório do Projeto de SO**

Rodolfo Silva (a81716)      Pedro Gonçalves (a82313)

Maio 2018

Universidade do Minho  
Sistemas Operativos  
Grupo 70

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Estratégias de Resolução dos Objetivos do Projecto</b>	<b>3</b>
2.1	Execução de Programas . . . . .	3
2.2	Re-Processamento de um Notebook . . . . .	3
2.3	Detecção de erros e interrupção da execução . . . . .	3
<b>3</b>	<b>Conclusão</b>	<b>4</b>

## 1 Introdução

Este relatório aborda a resolução do projeto prático de SO. O projeto consiste em construir um sistema para processamento de notebooks (ficheiros de texto que depois de processados são modificados de modo a incorporar os resultados do código/comandos neles embetidos).

## 2 Estratégias de Resolução dos Objetivos do Projecto

### 2.1 Execução de Programas

Para resolver este primeiro objectivo, o grupo optou por ler o ficheiro passado como argumento linha a linha, usando para isso a função *readln*, que sempre que detetava um caracter diferente do caracter dólar no início de uma linha, escrevíamos essa mesma linha no ficheiro temporário. Caso o caracter inicial fosse o dólar, iríamos analisar o segundo caracter da linha. Visto que o grupo decidiu realizar a primeira funcionalidade avançada do projeto, para esse segundo caracter da linha iremos ter três possibilidades (espaço vazio, barra ou um número), e para todos estes casos, percorreríamos o buffer que continha essa linha para obter o comando a executar e os argumentos desse mesmo comando. Se o segundo caracter for uma barra ou um número iríamos utilizar o output do comando anteriormente executado ou do n-ésimo comando, utilizando para isso a função *lseek* que fornece os descritores de ficheiros do início e do fim do output de um certo comando, abrindo depois o ficheiro temporário escrevíamos num pipe o output do comando anterior que ia depois ser o stdin do comando que pretendíamos executar.

### 2.2 Re-Processamento de um Notebook

Na resolução do segundo objectivo, o grupo optou por uma implementação simples que sempre que ao ler uma linha em que o caracter `>` for detetado no início da linha uma flag vai ser ativada que indica ao programa que enquanto essa mesma flag está ativa o programa não irá escrever nada no ficheiro temporário, o que resulta na não existência de outputs duplicados. Esta flag vai voltar a ter um valor que permita a escrita no ficheiro temporário assim que o caracter `<` for detetado no início de uma linha.

### 2.3 Detecção de erros e interrupção da execução

Para a resolução do objetivo final, foi utilizado a função *signal* que é chamada no início do programa e que vai esperar pelo input de um Control+C, e quando esse input for detetado vai imprimir uma mensagem no terminal a confirmar que apanhou o input e consecutivamente vai fechar o programa. De resto sempre que se utilizavam forks o pai espera que o filho acaba de executar os exec e

caso o valor do exit do filho fosse diferente de 0, significa que algo de errado aconteceu no exec e consequentemente o pai vai dar shutdown ao programa e não vão ocorrer alterações ao notebook fornecido.

### **3 Conclusão**

Em suma, concluímos o projeto com uma sensação de dever cumprido. Aumentamos os nossos conhecimentos na disciplina de Sistemas Operativos e consideramos que o projecto foi uma ajuda ao estudo para o teste desta cadeira.