



**POLITÉCNICO
DE LEIRIA**

ESCOLA SUPERIOR
DE TECNOLOGIA
E GESTÃO

Relatório do Projeto da
Unidade Curricular de
Sistemas Operativos

2020-21

Script *Bash*

Simão Pereira Troeira



Nº: 2201701

Flávio Pereira Da Costa



Nº: 2201707

“Simão Pereira Troeira (2201701) e por Flávio Pereira Da Costa (2201707) declaram sob compromisso de honra que o presente trabalho (código, relatórios e afins) foi integralmente realizado por nós, sendo que as contribuições externas se encontram claramente e inequivocamente identificadas no próprio código. Mais se declara que os estudantes acima identificados não disponibilizaram o código ou partes dele a terceiros”.

Estado dos Modos:

• Modos c/C: Totalmente Operacionais

• Modos p/P: Totalmente Operacionais

• Modos t/T: Totalmente Operacionais

Implementação de funcionalidades:

- A função **func_INPUT()** recebe o conteúdo a processar (ficheiro de texto ou pdf, sendo que o pdf deve ser convertido em texto). Além do mais, a função também verifica se o ficheiro é inválido (caso não seja pdf ou texto) e se existe.
- A função **func_ISO3166()** verifica o idioma em que se encontra o conteúdo especificado pelo parâmetro da linha de comando <INPUT>. A funcionalidade só suporta valores 'pt' e 'en', respetivamente para português e inglês. O parâmetro <ISO3166> é opcional, caso não seja especificado ou não tenha um valor válido, deve-se assumir que o conteúdo se encontra em inglês.
- A função **func_top_words()** processa o valor da variável ambiente WORD_STATS_TOP, caso essa variável não se encontre definida ou não corresponda a um número, é assumido o valor 10.
- A função **func_Pp()** encarrega-se de construir o gráfico e criar dois ficheiros, uma imagem em png e um ficheiro html. O ficheiro png é aberto automaticamente após o processamento.

Metodologia de Contagem:

```
grep -oE '[:alpha:]]+' $INPUT
```

- Filtra todas as palavras que vieram do parâmetro <INPUT> que contem caracteres alfabéticos e separa-as por linhas.

```
grep -vwFf $STOP_WORDS_FILE
```

- Caso o modo seja minúsculo, este comando vai retirar as *stopwords* à filtragem do comando anterior.

```
sort | uniq -c | sort -nr
```

- Na primeira *pipe*, o comando **sort** vai ordenar as palavras por ordem alfabética, posteriormente na segunda *pipe* o comando **uniq -c** vai contar quantas vezes se repete cada palavra. Por fim o comando **sort -nr** ordena as palavras por ordem decrescente de ocorrência.

```
sed -n 1,$WORD_STATS_TOP p
```

- Este comando só mostra o *top* de N palavras que ocorrem mais vezes definida pela variável `WORD_STATS_TOP`.

```
awk '{print NR, $0}'
```

- Este comando enumera por ordem crescente o *ranking* de cada palavra.

```
cat > result---$INPUT
```

- Através deste comando é guardado o processo dos comandos anteriores e gravado num ficheiro.

```
tee result---$INPUT
```

- Este comando, similarmente ao referido anteriormente, grava num ficheiro, porém também mostra as informações na linha de comandos.

Conclusão:

Na nossa perspetiva, o projeto foi concluído na íntegra e com todas as otimizações necessárias. As verificações para o funcionamento correto do script também foram implementadas, de acordo com o que era pedido.

Durante a realização, recorreu-se ao esclarecimento de dúvidas por email com o das folhas das aulas práticas e da internet para o desenvolvimento de soluções de erros e simplificação de código.

Web grafia:

- <https://devhints.io/bash>
- http://www.gnuplot.info/docs_5.4/Gnuplot_5_4.pdf
- <https://www.gnu.org/software/grep/manual/grep.html#General-Output-Control>
- <https://man7.org/linux/man-pages/man1/>
- <https://stackoverflow.com/questions/57181863/create-html-page-from-within-bash-script>
- Fichas Práticas N°1 e 2