Sistemas Operativos

Projeto - EUGÊNIO

Ano: 2019/2020 / 1º semestre / 2º Ano

O Docente:
Luís B Garcia

Discentes:
Jorge Colaço/16524
Hugo Alexandre Silva/18544

Beja, aos 25 de Novembro de 2019

Índice

1. Introdução

Este projeto roda em desenvolvimento de scripts para a matéria de Sistemas Operacionais onde o professor responsável Luiz Garcia nos deu como tarefa criar um conjunto de scripts que permitam a criação de dicionários do Eugénio V3 para a língua inglesa.

Além disso os scripts devem fornecer informações sobre os textos utilizados (corpus) e os dicionários criados, como por exemplo o seu número de palavras, o tamanho médio das palavras, entre outras estatísticas. O corpus utilizado neste trabalho será o SwitchBoard. Este corpus contém um conjunto de chamadas telefónicas em inglês, transcritas para ficheiros de texto.

Com isso o objetivo do projeto é utilizar estes scripts para serem utilizados e incorporados junto aos ficheiros do SwitchBoard e utilização junto com o aplicativo Eugénio.

Para iniciarmos as atividades seguimos com as seguintes ações:

1- Criamos a pasta “tg1” onde teremos os ficheiros dos scripts, juntamente com o ficheiro SwitchBoard, conforme imagem1.

2- O script “principal” contempla a escrita dos script’s 1,2,3 e 4 dentro do seu corpo e assim como sua sequência de execução.

3- Todos os script’s serão executados em forma de chamadas, bastando o utilizador ou tecnico responsavel executar o script “principal”.

4- Observação para ter instalado na maquina os pacotes do gawk. Comando para instalação “sudo apt-get install gawk”.

5- O utilizador não precisará efetuar comandos de permissões para execução dos script’s pois isso já se encontra pervisto dentro das linhas de comandos e execução no script “principal”.

6- O script “principal” contempla a execução de cópia dos script’s 2 em diante para dentro do diretório script ao serem executados, deixando os mesmos armazenados neste diretório.

Concluímos que se tudo ocorrer devidamente correto teremos dentro das pastas os devidos ficheiros criados conforme enunciados e com os devidos conteúdos.

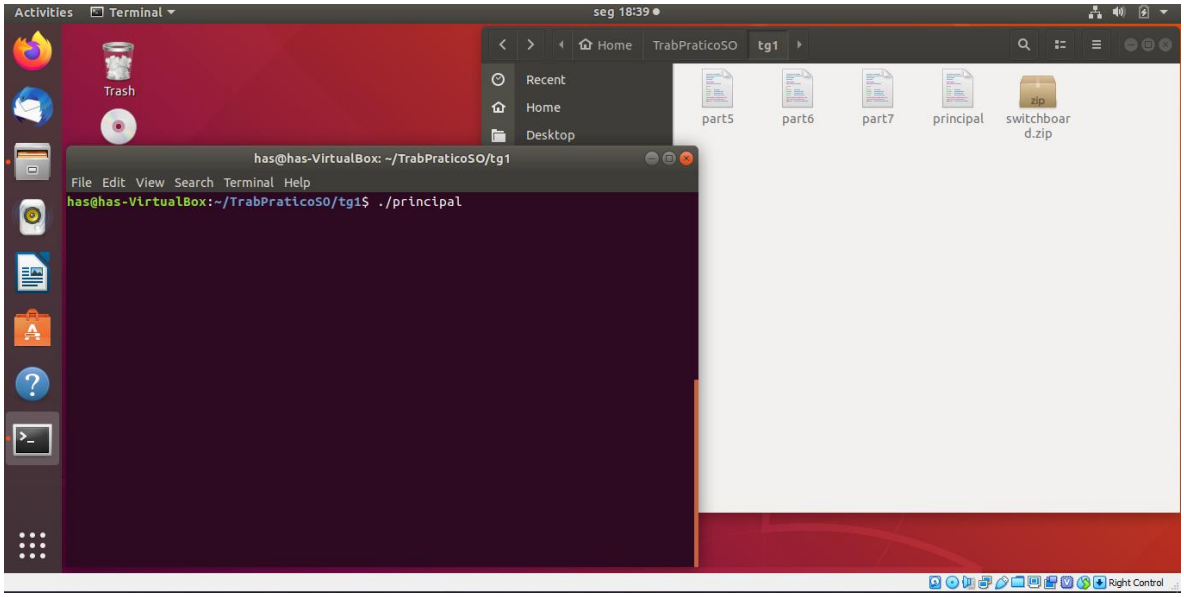

Imagem1 – execução do script principal

2. Criação de diretórios e subdiretórios

Crie uma diretoria tg1, e dentro desta, crie as seguintes subdiretorias: scripts, corpus, corpus_txt, corpus_info, words_dic, sentences_dic.

Baseando neste problema criamos o script para criação das diretórios e subdiretórios conforme script:

```
#!/bin/bash
if [ ! -d "./tg1" ]; then
mkdir ./tg1
mkdir ./tg1/scripts
mkdir ./tg1/corpus
mkdir ./tg1/corpus_txt
mkdir ./tg1/corpus_info
mkdir ./tg1/words_dic
mkdir ./tg1/sentences_dic
echo "Directories created with success."
```

Conforme demonstra a imagem2 abaixo ao ser executado o ficheiro “principal” o script1 de criação dos diretórios é acionado automaticamente pelo chamamento, isso após a confirmação de palavra chave do utilizador para permissões de execução. Chamamos o primeiro diretório como tg1 conforme imagem2 e os demais diretórios vem em sequência dentro desta tg1.

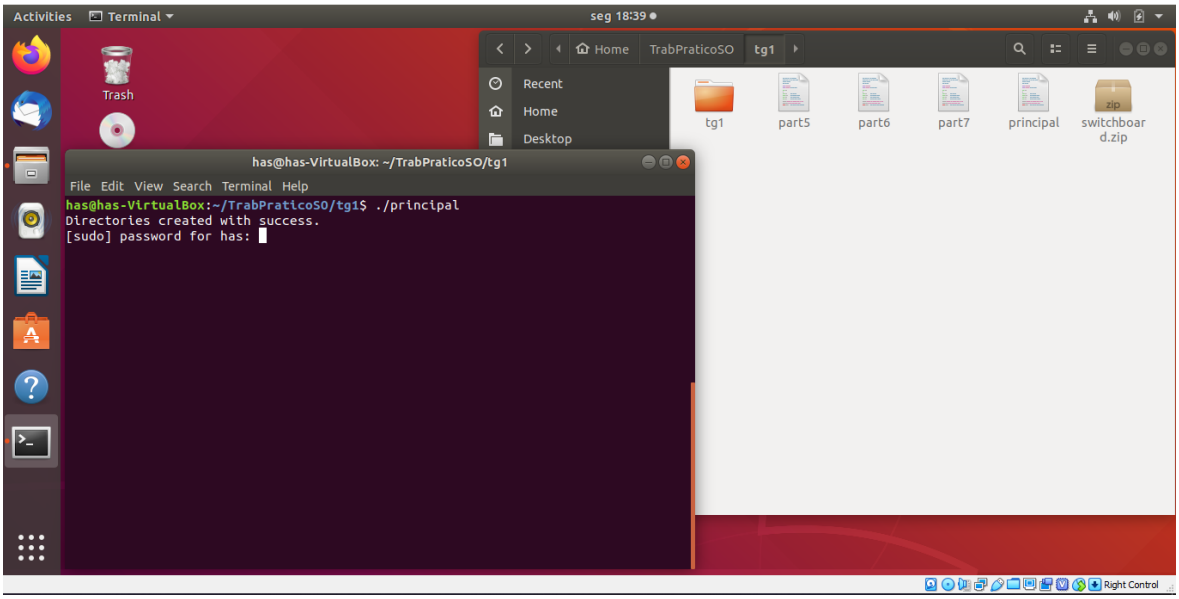

Imagem2 – pedido de chave de segurança e execução script1

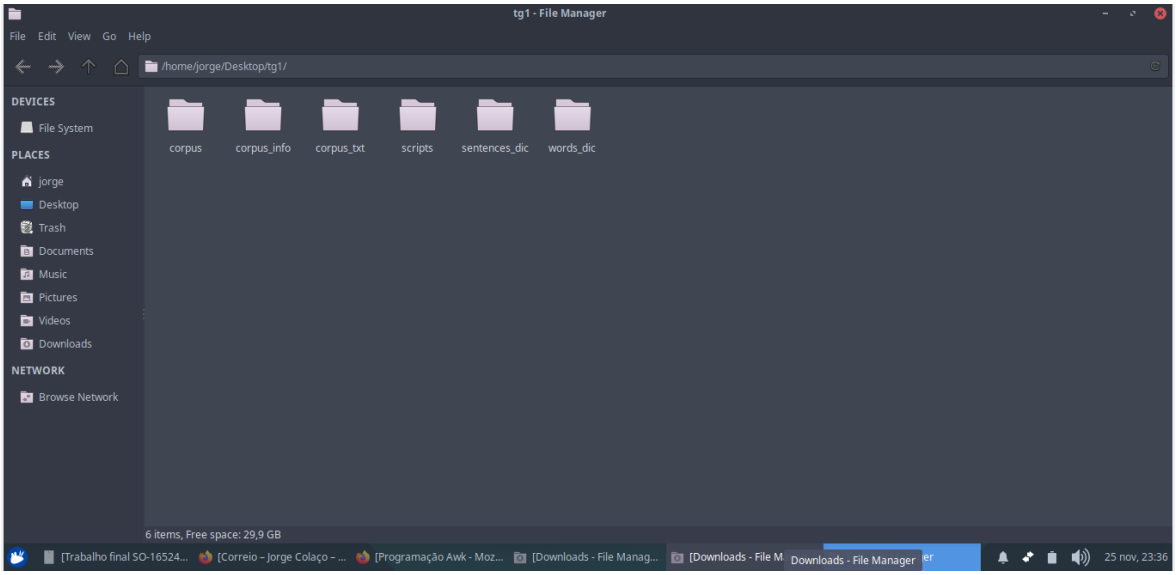

Imagem7 – subdiretórios

3. Junção de ficheiros

Descompacte o corpus SwitchBoard na diretoria corpus. Como pode verificar, além dos ficheiros de texto com as transcrições das conversas telefónicas, este corpus contém outros ficheiros relacionados, que fornecem outra informação sobre essas conversas telefónicas. Como apenas irá utilizar os ficheiros com as transcrições, junte num só ficheiro todos os ficheiros .txt deste corpus. Este ficheiro deverá ser armazenado na subdiretoria corpus_txt. O script que realiza esta operação deve ser armazenado na diretoria scripts.

Baseado neste problema pensamos na possibilidade de uso de dois extratores de ficheiros, o engrampa ou unzip, dependendo das versões Linux, poderá ter os dois ou apenas um dos, sendo assim já estamos na preparação desse quesito. Após conclusão das extrações o direcionamento para criação de um ficheiro único é direcionado e será criado com o seguinte nome: “swmerged-ms98-a-trans.txt”, conforme parte código abaixo e imagem3.

```
#!/bin/bash
echo "Attempting file extraction, checking if there's a file manager program compatible with .zip files."
if [ -x "$(command -v ark)" ]; then  echo "Engrampa exists, proceding with file extraction."
engrampa -e ./tg1/corpus/ ./switchboard.zip elif [ -x "/usr/bin/unzip" ]; then
echo "Unzip exists, proceding with file extraction."    unzip ./switchboard.zip -d ./tg1/corpus/ fi
find ./tg1/corpus/ -type f -name '*.txt' -exec cat > ./tg1/corpus_txt/swmerged-ms98-a-trans.txt {} +
ENDOFFILE
sudo chmod u+rwx ./tg1/scripts/part2
./tg1/scripts/part2 echo "2nd script executed with no errors."
```

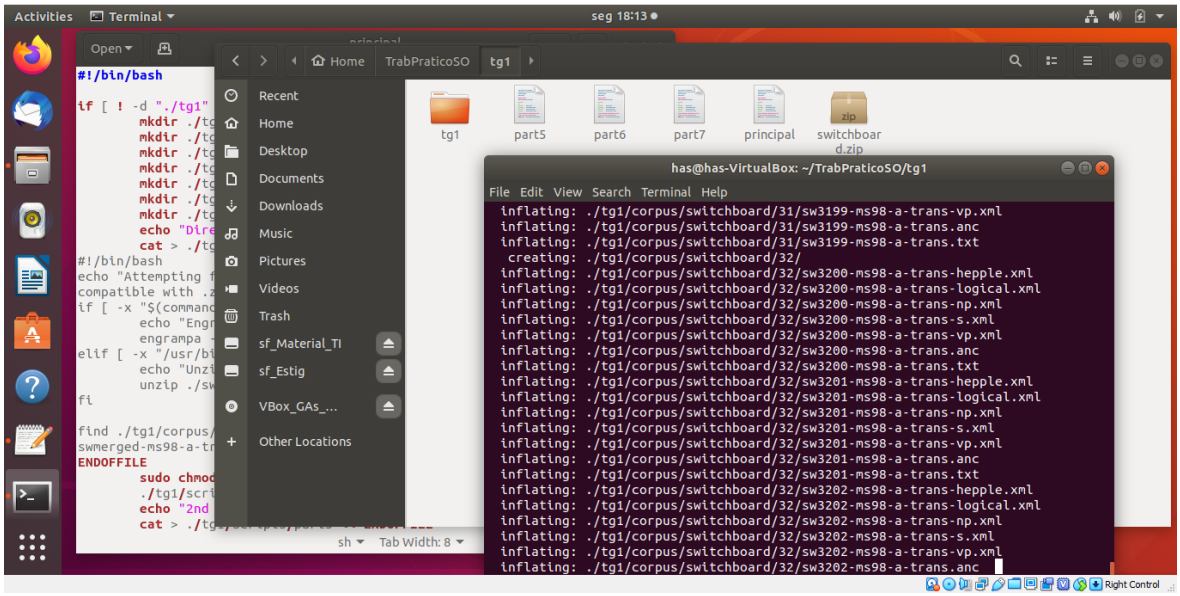

Imagem3 – extração dos ficheiros e criação do ficheiro único

4. Informações de caracteres, palavras, frases

Para caracterizar o corpus utilizado (switchboard) desenvolva um script que calcula as seguintes medidas: número de caracteres, quantidade de linhas não vazias, número total de palavras, número total de palavras diferentes, o quociente entre o total de palavras diferentes e o total de palavras, número total de frases, o número total de frases diferentes, o quociente entre o total de frases diferentes e o total de frases. O script que realiza esta operação deve

ser armazenado na diretoria scripts. O resultado do script deve ser armazenado na subdiretoria corpus_info, num ficheiro denominado corpus_info.txt.

Para resolução desse enunciado utilizamos “sed” que é um editor de textos não interativo. Vem do inglês [S]tream [ED]itor, ou seja, editor de fluxos de texto.

O comando cat que é uma derivação da palavra concatenate (concatenar) e permite que você crie, una e exiba arquivos no formato padrão de tela ou em outro arquivo, entre outras coisas. Além de comandos sequenciais para cada caso conforme demonstra a imagem4 e conteúdo do “dicionário significado” na página 15 deste documento.

```
#!/bin/bash
echo "Charater Nº: " > ./tg1/corpus_info/corpus_info.txt
wc -m < ./tg1/corpus_txt/swmerged-ms98-a-trans.txt >> ./tg1/corpus_info/corpus_info.txt
echo "Line Nº: " >> ./tg1/corpus_info/corpus_info.txt
cat ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | sed '/^\s*$/d' | wc -l >>
./tg1/corpus_info/corpus_info.txt  echo "Word Nº: " >> ./tg1/corpus_info/corpus_info.txt
wc -w < ./tg1/corpus_txt/swmerged-ms98-a-trans.txt >> ./tg1/corpus_info/corpus_info.txt
echo "Unique word Nº: " >> ./tg1/corpus_info/corpus_info.txt
cat ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | egrep -o "(\w|')+" | sort -f | uniq -i | wc -l >>
./tg1/corpus_info/corpus_info.txt  echo "Sentence Nº: " >> ./tg1/corpus_info/corpus_info.txt
sed 's/ \t\t//g' ./tg1/corpus_txt/swmerged-ms98-a-trans.txt > ./tg1/corpus_info/swmerged-ms98-a-
trans-temp.txt  sed 's/ \t//g' ./tg1/corpus_info/swmerged-ms98-a-trans-temp.txt >
./tg1/corpus_info/swmerged-ms98-a-trans-temp2.txt  sed 's/\.\. /g' ./tg1/corpus_info/swmerged-ms98-a-
trans-temp2.txt > ./tg1/corpus_info/swmerged-ms98-a-trans-temp3.txt
sed 's/\!/\! /g' ./tg1/corpus_info/swmerged-ms98-a-trans-temp3.txt > ./tg1/corpus_info/swmerged-
ms98-a-trans-temp4.txt  sed 's/\?/\? /g' ./tg1/corpus_info/swmerged-ms98-a-trans-temp4.txt >
./tg1/corpus_info/swmerged-ms98-a-trans-temp5.txt  sed 's/\;/\; /g' ./tg1/corpus_info/swmerged-ms98-a-
trans-temp5.txt > ./tg1/corpus_info/swmerged-ms98-a-trans-temp6.txt  sed 's/\. /\.\n/g'
./tg1/corpus_info/swmerged-ms98-a-trans-temp6.txt > ./tg1/corpus_info/swmerged-ms98-a-trans-temp7.txt
sed 's/\! /\!\n/g' ./tg1/corpus_info/swmerged-ms98-a-trans-temp7.txt > ./tg1/corpus_info/swmerged-
ms98-a-trans-temp8.txt  sed 's/\? /\?\n/g' ./tg1/corpus_info/swmerged-ms98-a-trans-temp8.txt >
./tg1/corpus_info/swmerged-ms98-a-trans-temp9.txt  sed 's/\; /\;\n/g' ./tg1/corpus_info/swmerged-ms98-a-
trans-temp9.txt > ./tg1/corpus_info/swmerged-ms98-a-trans-temp10.txt
wc -l < ./tg1/corpus_info/swmerged-ms98-a-trans-temp10.txt >> ./tg1/corpus_info/corpus_info.txt
echo "Unique entence Nº: " >> ./tg1/corpus_info/corpus_info.txt
cat ./tg1/corpus_info/swmerged-ms98-a-trans-temp10.txt | sort -f | uniq -i | wc -l >>
./tg1/corpus_info/corpus_info.txt  rm ./tg1/corpus_info/swmerged-ms98-a-trans-temp*.txt
ENDOFFILE
sudo chmod u+rwx ./tg1/scripts/part3  ./tg1/scripts/part3  echo "3rd script executed with no errors."
```

5. Ficheiro de palavras

Desenvolva um script que cria o ficheiro de palavras. Este ficheiro deverá conter em cada linha, uma palavra e as suas ocorrências no corpus. O ficheiro deverá ficar ordenado de forma alfabética. O script que realiza esta operação deve ser armazenado na diretoria scripts. O resultado do script deve ser armazenado na subdiretoria words_dic, num ficheiro denominado words.txt.

Nesta parte do script executamos através de cat e sed como já citados acima e o comando egrep que procura um padrão de texto, usando expressões regulares estendidas para executar a correspondência. Como complemento da ação dos scripts seguinte, deixamos descrito nesta parte do script que está incorporado dentro do "principal" o chamamento, execução e copia dos scripts para dentro da pasta script, conforme código abaixo e imagem4 e na imagem5 a criação dos ficheiros scripts.

```
#!/bin/bash
cat ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | egrep -o '\w+' | sort -f | uniq -c -i >
./tg1/words_dic/words.txt ENDOFFILE
sudo chmod u+rwx ./tg1/scripts/part4 ./tg1/scripts/part4
echo "4th script executed with no errors." cp ./part5 ./tg1/scripts/
sudo chmod u+rwx ./tg1/scripts/part5 ./tg1/scripts/part5
echo "5rd script executed with no errors." cp ./part6 ./tg1/scripts/
sudo chmod u+rwx ./tg1/scripts/part6 ./tg1/scripts/part6
echo "6rd script executed with no errors." cp ./part7 ./tg1/scripts/
sudo chmod u+rwx ./tg1/scripts/part7 ./tg1/scripts/part7
echo "7rd script executed with no errors." cp ./part8 ./tg1/scripts/
sudo chmod u+rwx ./tg1/scripts/part8
./tg1/scripts/part8 echo "8th script executed with no errors." else
echo "Directory tg1 already exists, aborting operation."l exit 1
fi
```

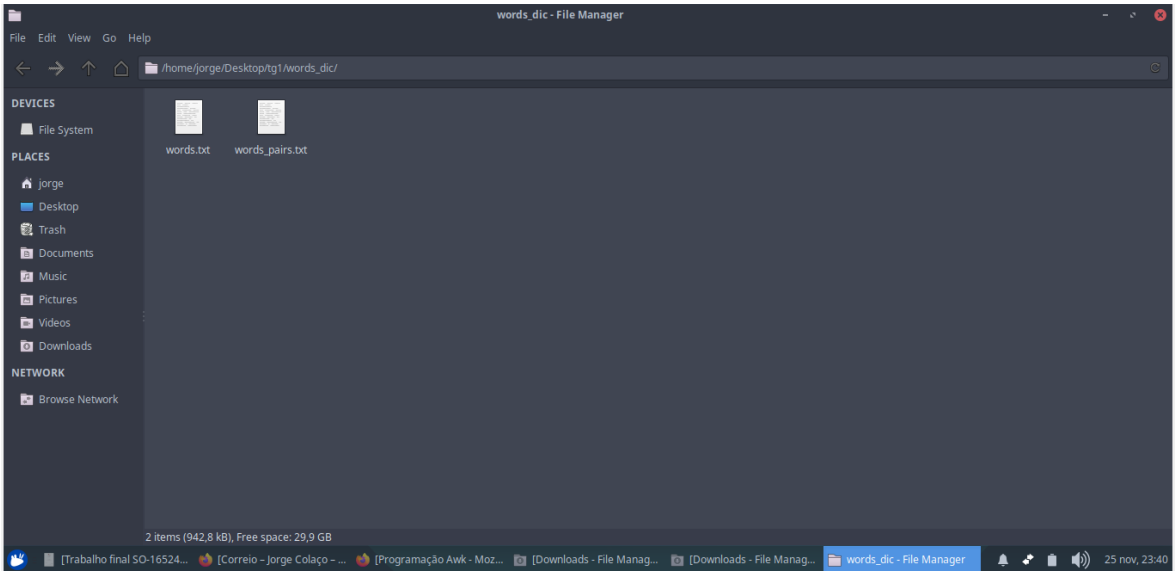

Imagem4 – palavras e palavras pares

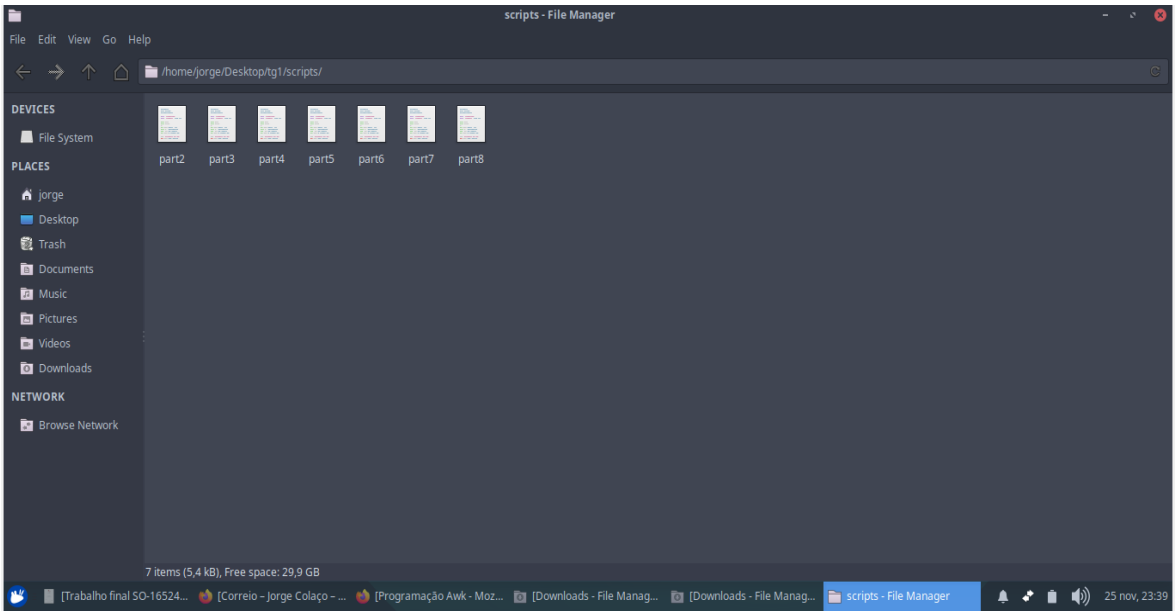

Imagem5 – scripts diretório script

6. Ficheiro pares de palavras

Desenvolva um script que cria o ficheiro com os pares de palavras. Este ficheiro deverá conter em cada linha um par de palavras e as suas ocorrências no corpus. O ficheiro deverá ficar ordenado de forma alfabética. O script que realiza esta operação deve ser armazenado na diretoria scripts. O resultado do script deve ser armazenado na subdiretoria words_dic, num ficheiro denominado words_pairs.txt.

Enunciado executado conforme parte do código abaixo e parâmetros já citados acima como sed e demais utilizando também gawk que é usado para verificação de padrões e linguagem de processamento e demonstrado a criação do ficheiro na imagem4 acima.

```
#!/bin/bash
sed -e 's/[^[:alpha:]]/ /g' ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | sort -f | gawk -F, '{
for(i=0; i <=NF; i++){j= i+1;
a[$i,$j]=$i" "$j; print a[$i,$j]; }
}' | tr " " "#" | sort -d | uniq -c | tr "#" " " | gawk '{print $2 " " $3}' | sort -f | uniq -c -i >
./tg1/words_dic/words_pairs.txt
```

7. Ficheiro de frases

Desenvolva um script que cria o ficheiro de frases. Este ficheiro deverá conter em cada linha uma frase e as suas ocorrências no corpus. Em cada frase, os espaços entre as palavras deverão ser substituídos pelo caractere '|'. O ficheiro deverá ficar ordenado de forma alfabética. O script que realiza esta operação deve ser armazenado na diretoria scripts. O resultado do script deve ser armazenado na subdiretoria sentences_dic, num ficheiro denominado sentences.txt.

Enunciado executado conforme parte do código abaixo e parâmetros já citados acima como sed e demonstrado a criação do ficheiro na imagem6 abaixo.

```
#!/bin/bash
sed -e 's/[^[:alpha:]]/ /g' ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | tr 'A-Z' 'a-z' | tr " " "|" | sort -d
| uniq -c | gawk '{ for(i = 0; i < NF; i++) {  print $i ""; }}' >> ./tg1/sentences_dic/sentences.txt
```

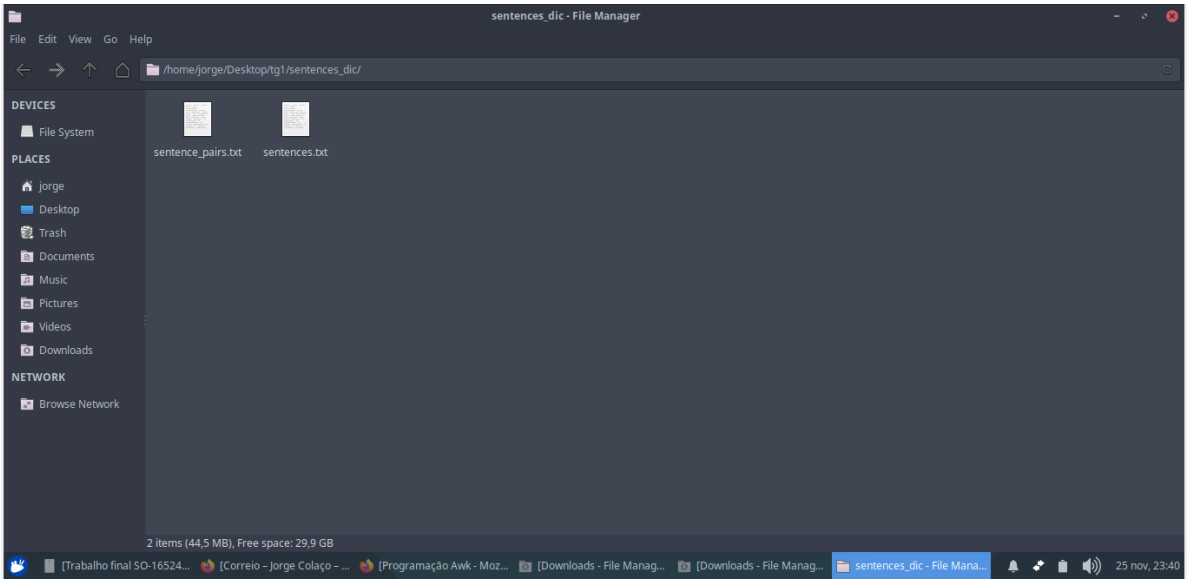

Imagem6 – frases e pares de frases

8. Ficheiro pares de frases

Desenvolva um script que cria o ficheiro com os pares de frases. Este ficheiro deverá conter em cada linha um par de frases e as suas ocorrências no corpus. Em cada frase, os espaços entre as palavras deverão ser substituídos pelo caractere '|'. O ficheiro deverá ficar ordenado de forma alfabética. O script que realiza esta operação dever ser armazenado na diretoria scripts. O resultado do script deve ser armazenado na subdiretoria sentences_dic, num ficheiro denominado sentences_pairs.txt.

Enunciado executado conforme parte do código abaixo e parâmetros já citados acima como sed e gawk, complemento de descrição do código está referido no dicionário significado pagina 15 demonstrado a criação do ficheiro na imagem6 acima.

```
#!/bin/bash
sed -e 's/[^[:alpha:]]/ /g' ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | sort -f | gawk
-F, '{for(i=0; i <NF; i++){ j= i+1; a[$i,$j]=$i" "$j; print a[$i,$j];}
}' | sort -d | uniq -c | tr " " "|" | sed -e 's/||/ /g' >
./tg1/sentences_dic/sentence_pairs.txt
```

9. Ficheiro 250.000 palavras

O Eugénio suporta apenas um máximo de 250.000 palavras e 250.000 frases. Através da modificação dos scripts anteriores, ou através de novos scripts, garanta que não existem no ficheiro mais de 250.000 palavras e frases.

```
#!/bin/bash
cat ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | egrep -o '\w+' | sort -f | uniq -c -i >
./tg1/words_dic/words_8.txt sed -n '3,250002p' ./tg1/words_dic/words_8.txt >
./tg1/words_dic/words_8_temp.txt  mv ./tg1/words_dic/words_8_temp.txt ./tg1/words_dic/words.txt
rm ./tg1/words_dic/words_8.txt
sed -e 's/[^[:alpha:]]/ /g' ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | sort -f | gawk -F, '{ for(i=0; i
<=NF; i++){ j= i+1; a[$i,$j]=$i" "$j; print a[$i,$j]; }
}' | tr " " "#" | sort -d | uniq -c | tr "#" " " | gawk '{print $2 " " $3;}' | sort -f | uniq -c -i >
./tg1/words_dic/words_pairs_8.txt
sed -n '3,250002p' ./tg1/words_dic/words_pairs_8.txt > ./tg1/words_dic/words_pairs_8_temp.txt
mv ./tg1/words_dic/words_pairs_8_temp.txt ./tg1/words_dic/words_pairs.txt
rm ./tg1/words_dic/words_pairs_8.txt
sed -e 's/[^[:alpha:]]/ /g' ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | tr 'A-Z' 'a-z' | tr " " "|" | sort -d
| uniq -c | gawk ' { for(i = 0; i < NF; i++) { print $i ""; }}' >> ./tg1/sentences_dic/sentences_8.txt
sed -n '18,2500017p' ./tg1/sentences_dic/sentences_8.txt > ./tg1/sentences_dic/sentences_8_temp.txt
```

```
mv ./tg1/sentences_dic/sentences_8_temp.txt ./tg1/sentences_dic/sentences.txt
rm ./tg1/sentences_dic/sentences_8.txt
sed -e 's/[^[:alpha:]]/ /g' ./tg1/corpus_txt/swmerged-ms98-a-trans.txt | sort -f | gawk -F, '{ for(i=0; i
<NF; i++){ j= i+1; a[$i,$j]=$i" "$j; print a[$i,$j]; }
}' | sort -d | uniq -c | tr " " "|" | sed -e 's/||/ /g' > ./tg1/sentences_dic/sentence_pairs_8.txt
sed -n '10,250009p' ./tg1/sentences_dic/sentence_pairs_8.txt >
./tg1/sentences_dic/sentence_pairs_8_temp.txt
mv ./tg1/sentences_dic/sentence_pairs_8_temp.txt ./tg1/sentences_dic/sentence_pairs.txt
rm ./tg1/sentences_dic/sentence_pairs_8.txt
```

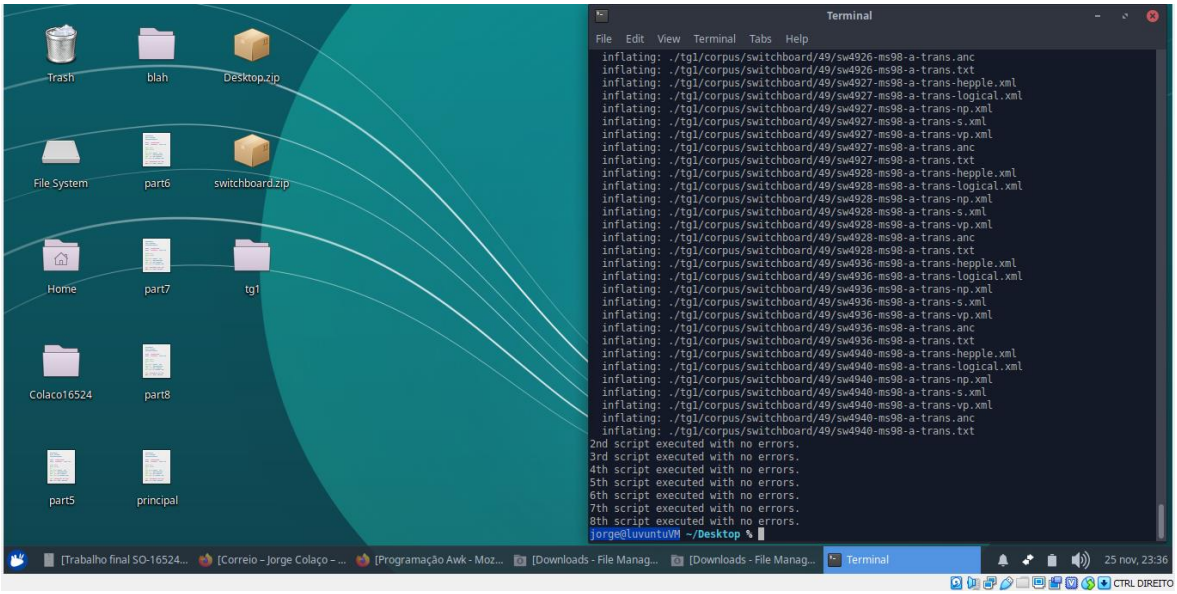

Imagem8 – restrição de 250.000

10. Script Windows

Desenvolva um script Windows que copie os ficheiros gerados para a diretoria do Eugénio. Se tiver instalado a versão 64 bits, a diretoria do Eugénio será C:\Program Files\Eugénio. No caso de ter instalado a versão de 32 bits a diretoria será C:\Program Files (x86)\ Eugénio. Para a instalação dos dicionários em inglês deve realizar as seguintes copias: • words.txt -> geral.pal • words_pairs.txt -> geral.par • sentences.txt –> geral.frs • sentences_pairs.txt -> geral.paf Para experimentar o script deve desinstalar o Eugénio, e voltar a correr a instalação. Antes de chamar o Eugénio deve correr o script que altera os dicionários. Só depois deve chamar o Eugénio.

11. Conclusões Finais

O trabalho foi apresentado e elaborado dentro da proposta de vários scripts, para ação e tarefas baseados a partir de um aplicativo chamado Eugénio. No decorrer do processo viu-se a necessidade de facilitar a execução dos scripts em quesitos de usabilidade e dinamicidade, afim de manter essa facilidade ao nível do utilizador, executamos todos os scripts a partir do script “principal”.

A elaboração de criação dos scripts demanda principalmente dos cuidados das permissões de acesso e escrita, usabilidade estrutural da criação dos diretórios e ficheiros, possibilitando a melhor adaptação e eliminando falhas para não prejudicar a execução das tarefas.

Baseado no proposto, podemos dizer que este trabalho foi um bom começo para o aprendizado sobre como trabalhar com SHELL, comandos, scripts e as facilidades além da segurança que o mesmo nos fornece no processo, assim sendo, o que nos incentivou este desenvolvimento sequêncial foi a possibilidade de ser aplicado em algo real e que pode ser utilizado para outros modelos e parâmetros.

Concluindo este projeto podemos dizer que a necessidade de um processo com respostas de confirmação em cada etapa dos scripts conforme descrito nos códigos, exemplo: “echo "2nd script executed with no errors.", traz a confiabilidade na execução, assim como a visualização dos resultados sólidos dos scripts, servem para saber se as demandas estão sendo efetuadas conforme enunciado e com as devidas respostas nas etapas.

Estamos satisfeitos com os resultados e conhecimento adquirido e esperamos ter alcançado os objetivos do professor!

12. Referências bibliográficas

Moodle

https://w1fultva6vqpc1zypcouvq-on.drv.tw/html/prog_awk.htm

https://w1fultva6vqpc1zypcouvq-on.drv.tw/html/prog_she.htm

https://w1fultva6vqpc1zypcouvq-on.drv.tw/html/re_pi_fi.htm

https://w1fultva6vqpc1zypcouvq-on.drv.tw/html/com_proc.htm

Youtube

Aurelio Jargas – sequencia de 9 videos de instrução

https://www.youtube.com/watch?v=BYYyt6Ag3Kc&list=PLkMH2SrZj2aiWw-t6rLgciBQqqoZZn5t1

Tutoriais

https://terminalroot.com.br/2015/07/30-exemplos-do-comando-sed-com-regex.html

https://www.livrosdelinux.com.br/respostas-livro-shell-linux/

13. Dicionário significado

É case sensitiva

- **-i** altera o arquivo
- **-e** imprime na tela sem alterar o arquivo
- **-n** faz a supressão, mostra só o resultado do comando
- **s** substitui um trecho de texto por outro
- **!** inverte a lógica do comando
- **;** separador de comandos
- | separador de strings
- **d** no final deleta
- **p** no final imprime
- **g** no final (como se usa o d e p) altera todas as ocorrências
- **q** sai do sed , não continua o comando

Classes de caracteres do GNU sed

- **[[:alnum:]]** Alfabéticos e númericos [a-z A-Z 0-9]
- **[[:alpha:]]** Alfabéticos [a-z A-Z]
- **[[:blank:]]** Caractere em branco, espaço ou tab [\t]
- **[[:cntrl:]]** Caracteres de controle [\x00-\x1F\x7F]
- **[[:digit:]]** Números [0-9]
- **[[:graph:]]** Qualquer caractere visível(ou seja, exceto em branco) [\x20-\x7E]
- **[[:lower:]]** Letras minúsculas [a-z]
- **[[:upper:]]** Letras maiúsculas [A-Z]
- **[[:print:]]** Caracteres visíveis (ou seja, exceto os de controle) [\x20-\x7E]
- **[[:punct:]]** Pontuação [-!”#$%&’()*+,./:;?@[\]_`{
- **[[:space:]]** Espaço em branco [\t\r\n\v\f]
- **[[:xdigit:]]** Número hexadecimais [0-9 a-f A-F]

O comando gawk pode ser usado para:

- Digitaliza um arquivo linha por linha.
- Divide cada linha de entrada em campos.
- Compara a linha / campos de entrada com o padrão.
- Executa ações nas linhas correspondentes.
- Transforme arquivos de dados.
- Produza relatórios formatados.
- Formate linhas de saída.
- Operações aritméticas e de cordas.
- Condicionais e loops.