

Processamento de Linguagens e Compiladores
(3º ano de LCC)

Trabalho Prático N°2 (GAWK)

2.3 Processar as Pessoas listadas nos Róis de Confessados com o Gawk
Relatório de Desenvolvimento

João Ferreira
(A76628)

Luís Daniel Félix
(A74246)

Maria Manuela Silva
(A74408)

14 de Janeiro de 2019

Conteúdo

1	Introdução	2
2	Análise e Especificação	3
2.1	Descrição informal do problema	3
2.2	Especificação do Requisitos	3
2.2.1	Dados	3
2.2.2	Pedidos	3
3	Conceção/desenho da Resolução	4
4	Codificação e Testes	5
4.1	Alternativas, Decisões e Problemas de Implementação	5
4.2	Testes realizados e Resultados	5
4.2.1	Execução do programa e Resultados obtidos	8
5	Conclusão	11
A	Código do Programa	12

Capítulo 1

Introdução

O trabalho prático 2.3, Processar as Pessoas listadas nos Róis de Confessados com o Gawk, surge inserido no âmbito da disciplina de *Processamento de Linguagens e Compiladores*. É o segundo trabalho prático desta unidade curricular, abordando o uso da ferramenta **GAWK**. Ao longo deste documento podemos encontrar a exposição do problema, a sua análise, a forma como o interpretamos e resolvemos e as dificuldades que surgiram ao longo da sua realização.

Capítulo 2

Análise e Especificação

2.1 Descrição informal do problema

Foi-nos fornecido um ficheiro em anexo de nome "*processos.txt*", que contém processos das pessoas listadas nos Róis de confessados. Pretende-se desenvolver um Processador de Texto com o **GAWK** para ler um o ficheiro anexado, ou seja, o objetivo é contruir um ficheiro chamado *expprocessos.gawk* para processar o texto 'processos.txt com o intuito de calcular frequências de alguns elementos (a ideia é utilizar arrays associativos para o efeito).

2.2 Especificação do Requisitos

2.2.1 Dados

Para resolver o problema que nos foi proposto dividimos o ficheiro *expprocessos.gawk* em 3 para que fossem satisfeitos os pedidos. No ficheiro *processos.txt* temos diferentes tipos de dados. No primeiro campo temos o número do processo, no segundo campo tem a data do processo, nos restantes campos encontram-se os nomes das pessoas envolvidas no processo, tipos de realções entre elas, entre outras informações. Para acedermos aos varios campos e fazer os calculos pedidos, tivemos de recorrer a arrays.

2.2.2 Pedidos

Neste trabalho, foram-nos dadas 3 alíneas para resolver. Na alínea (a) é-nos pedido que calculemos a frequência de processos por ano (primeiro elemento da data); Na alínea (b) é-nos pedido que calculemos a frequência de nomes (considera um nome uma palavra e propaga o cálculo por todos os campos que contenham nomes); Na alínea (c) é-nos pedido que calculemos a frequência dos vários tipos de relação: irmão, sobrinho, etc.

Capítulo 3

Conceção/desenho da Resolução

Para a realização das 3 alneas, criamos 3 fcheiros, um para cada uma das alneas denominados de *exepprocessosa.gawk*, *exepprocessosb.gawk*, *exepprocessosc.gawk* respetivamente. Na alinea **a** e **b** vamos buscar o que desejamos aos campos queremos sem o uso de expresões regulares, recorrendo apenas a funções do gawk e da linguagem C, ja na alinea **c** recorreremos tambem ao uso de ER's (Expressões Regulares).

Capítulo 4

Codificação e Testes

4.1 Alternativas, Decisões e Problemas de Implementação

Depois de estudarmos o enunciado, fizemos em relativamente pouco tempo a as alíneas a e c. A nossa dificuldade prendeu-se no problema b uma vez que temos de imprimir todos os nomes e os mesmos do ultimo campo nao imprimem devido ao formato em que está o texto.

4.2 Testes realizados e Resultados

Na **alínea a**), usamos o *split* para passar o campo 2 (que é o campo que contem a data) para um array chamado *data*, sempre que encontra -. Seguidamente, é guardado cada ano (que se encontra na posição 1 do array data, data[1]) no array count e é feita a sua contagem.

```
BEGIN {FS = ":"}

    {
        split($2,data,"-");
        count[data[1]]++;
    }

END {
    for (ano in count) {print "No ano " ano " há " count[ano] " processos";}
}
```

Figura 4.1: Implementação da alínea a

Na **alinea b)** começamos por dividir os campos e começar a contar a frequência da nomes através de um array denominado *freq*. Seguiu-se um caso particular para o campo 6 onde se encontram campos estruturados de forma diferente. Devido a essa estrutura diferente, fomos tentando ir buscar os nomes existentes sendo que tivemos sucesso em alguns mas noutros não. A codificação da alinea b encontra-se de seguida na figura 4.2.

```
BEGIN{FS = "::"}
{
    for(i=3; i<NF-1; i++){
        {freq[$i]++};
    }
    {split($6,name,",");
    freq[name[1]]++;
    {split(name[1],name3,".")
    freq[name3[3]]++;}
    {split(name[2],name2,".");
    freq[name2[4]]++;}
    }
}

END {
    for (nome in freq) {print "O nome " nome " aparece " freq[nome] " vezes"}
}
```

Figura 4.2: Implementação da alinea b

Na **alinea c)** implementamos um ciclo para correr todas as linhas e campos do programa e porcurar as relações existentes através de expressões regulares, e incrementar de modo a dar o número de relações de cada tipo que de facto existem. A codificação da alinea c encontra-se nas duas imagens a baixo.

```
BEGIN{FS = "."}
{
  for(i=1; i<=NF; i++){
    if($i ~ /,Irmaos/){
      {rela["Irmãos"]++};

      if($i ~ /,Irmão/){
        {rela["Irmão"]++};

        if($i ~ /,Tio Paterno/){
          {rela["Tio Paterno"]++}

          if($i ~ /,Tio Materno/){
            {rela["Tio Materno"]++};

            if($i ~ /,Pai/){
              {rela["Pai"]++};

              if($i ~ /,Pais/){
                {rela["Pais"]++};

                if($i ~ /,Sobrinho Materno/){
                  {rela["Sobrinho Materno"]++};

                  if($i ~ /,Sobrinho Paterno/){
                    {rela["Sobrinho Paterno"]++};

                    if($i ~ /,Primo/){
                      {rela["Primo"]++};

                      if($i ~ /,Avo Materno/){
                        {rela["Avo Materno"]++};

                        if($i ~ /,Avo Paterno/){
                          {rela["Avo Paterno"]++};

                          if($i ~ /,Tio Avo Materno/){
                            {rela["Tio Avo Materno"]++};

                            if($i ~ /,Tio Avo Paterno/){
                              {rela["Tio Avo Paterno"]++};

                              if($i ~ /,Neto Paterno/){
                                {rela["Neto Paterno"]++};
```

Figura 4.3: Implementação da alinea c (parte 1)


```

46     if($i ~ /,Neto Paterno/)
47     {rela["Neto Paterno"]++};
48
49     if($i ~ /,Neto Materno/)
50     {rela["Neto Materno"]++};
51
52     if($i ~ /,Sobrinho Bisneto Paterno/)
53     {rela["Sobrinho Bisneto Paterno"]++};
54
55     if($i ~ /,Sobrinho Bisneto Materno/)
56     {rela["Sobrinho Bisneto Materno"]++};
57 }
58 }
59
60 END {
61     for (r in rela) {print "Da relação " r " há " rela[r] " relações"}
62 }
63

```

Figura 4.4: Implementação da alínea c (parte 2)

4.2.1 Execução do programa e Resultados obtidos

Para a execução do programa, criamos uma Makefile com os comandos necessários. Esta makefile compila os 3 ficheiros ao mesmo tempo.

```

Tarefas:
    gawk -f expprocessosa.gawk < processos.txt
    gawk -f expprocessosc.gawk < processos.txt
    gawk -f expprocessosb.gawk < processos.txt

```

Figura 4.5: Makefile

Depois de executarmos a makefile obtemos os resultados pretendidos para cada alínea, como se pode ver seguidamente nas imagens.

Como o ficheiro "processos.txt" é enorme, os resultados obtidos também são enormes e por isso não conseguimos colocar aqui imagens dos resultados completos, apenas de partes.

```

No ano 1868 há 22 processos
No ano 1869 há 39 processos
No ano 1870 há 12 processos
No ano 1871 há 35 processos
No ano 1872 há 42 processos
No ano 1873 há 46 processos
No ano 1874 há 46 processos
No ano 1875 há 15 processos
No ano 1876 há 40 processos
No ano 1877 há 47 processos
No ano 1878 há 76 processos
No ano 1879 há 55 processos
No ano 1880 há 62 processos
No ano 1881 há 64 processos
No ano 1882 há 50 processos
No ano 1883 há 34 processos
No ano 1884 há 44 processos
No ano 1885 há 8 processos
No ano 1886 há 46 processos
No ano 1887 há 42 processos
No ano 1888 há 70 processos
No ano 1889 há 71 processos
No ano 1890 há 44 processos
No ano 1891 há 73 processos
No ano 1892 há 57 processos
No ano 1893 há 78 processos
No ano 1894 há 74 processos
No ano 1895 há 78 processos
No ano 1896 há 75 processos
No ano 1897 há 71 processos
No ano 1898 há 91 processos
No ano 1899 há 80 processos
No ano 1900 há 50 processos
No ano 1901 há 59 processos
No ano 1902 há 83 processos
No ano 1903 há 70 processos

```

Figura 4.6: Resultado obtido da execução de "expprocessosa.gawk"

Nesta figura observa-se que obtivemos os anos existentes e respetivo número de processos nesse ano

```

0 nome Joana Maria Queiros aparece 1 vezes
0 nome Andresa Costa Erosa aparece 1 vezes
0 nome Baptizado com o nome de CAETANO aparece 1 vezes
0 nome Gregorio Alvares Oliveira aparece 1 vezes
0 nome Bernardino Gomes aparece 2 vezes
0 nome Maria Joana Rodrigues aparece 4 vezes
0 nome Margarida Rodrigues Soares aparece 1 vezes
0 nome Antonio Abreu Castro aparece 3 vezes
0 nome Pedro Alvares Sousa aparece 1 vezes
0 nome Baltazar Vieira aparece 1 vezes
0 nome Joao Luis Vieira aparece 2 vezes
0 nome Mateus Luis Costa Pinto aparece 3 vezes
0 nome Antonio Rodrigues Fontao aparece 1 vezes
0 nome Em Anexo: Inquiricao feita em 1702/02/17. Bento Araujo Correia aparece 1 vezes
0 nome Antonio Barbosa Gois aparece 3 vezes
0 nome Joao Batista Barao aparece 1 vezes
0 nome Diogo Jose Marques aparece 2 vezes
0 nome Em Anexo: Justificacao. Bento Rego Barbosa aparece 1 vezes
0 nome Margarida Mendes Pereira aparece 3 vezes
0 nome Joaquim Fernandes Santos aparece 2 vezes

```

Figura 4.7: Resultado obtido da execução de "expprocessosb.gawk"

Nesta figura 4.6 encontram-se os resultados obtidos após a execução do código. No entanto não é o resultado que esperávamos pois não conseguimos tirar frases que não correspondem a nomes.

```
Da relação Tio Avo Materno há 230 relações
Da relação Avo Materno há 48 relações
Da relação Neto Paterno há 8 relações
Da relação Sobrinho Bisneto Paterno há 3 relações
Da relação Pai há 525 relações
Da relação Sobrinho Materno há 1698 relações
Da relação Irmão há 14431 relações
Da relação Tio Materno há 2463 relações
Da relação Tio Avo Paterno há 154 relações
Da relação Avo Paterno há 11 relações
Da relação Sobrinho Paterno há 1642 relações
Da relação Primo há 1176 relações
Da relação Sobrinho Bisneto Materno há 3 relações
Da relação Neto Materno há 41 relações
Da relação Irmãos há 711 relações
Da relação Tio Paterno há 2245 relações
```

Figura 4.8: Resultado obtido da execução de "expprocessosc.gawk"

Nesta figura observa-se que obtivemos o número de relações para cada tipo de relação existente.

Capítulo 5

Conclusão

O (G)Awk é uma linguagem e processador de padrões em texto. A função básica do GAWK é processar linha a linha num determinado ficheiro de entrada que possua certos padrões especificados no programa. Para cada padrão deve haver uma ação associada, isto é, quando uma linha corresponde a um dos padrões, o GAWK realiza a ação correspondente naquela linha, caso contrário, ignora. Depois continua processando as linhas de entrada desta forma até encontrar o fim do ficheiro.

A realização deste trabalho permitiu-nos perceber melhor o funcionamento e a aplicação do GAWK. É um excelente filtro e processador de ficheiros de texto, especialmente quando queremos processar uma grande quantidade de informação, sendo mais fácil de usar do que a maioria das linguagens de programação convencionais.

De um modo geral fazemos um balanço positivo do nosso trabalho mesmo nao tendo concluido uma das alineas com sucesso.

Apêndice A

Código do Programa

Lista-se de seguida o código GAWK do programa que foi desenvolvido para a alínea a) do problema.

```
EGIN {FS = ":::"}

{
    split($2,data,"-");
    count[data[1]]++;
}

END {
    for (ano in count) {print "No ano " ano " há " count[ano] " processos";}
}
```

Lista-se agora o código GAWK do programa que foi desenvolvido para a alínea b) do problema.

```
BEGIN{FS = ":::"}
{
    for(i=3; i<NF-1; i++){
        {freq[$i]++};
    }
    {split($6,name,",");
    freq[name[1]]++;
    {split(name[1],name3,".")
    freq[name3[3]]++;}
    {split(name[2],name2,".")
    freq[name2[4]]++;}
    }
}

END {
    for (nome in freq) {print "O nome " nome " aparece " freq[nome] " vezes"}
}
```

Lista-se a baixo o código GAWK do programa que foi desenvolvido para a alínea c) do problema.

```
BEGIN{FS = "."}
{
    for(i=1; i<=NF; i++){
```

```

if($i ~ /,Irmaos/)
    {rela["Irmãos"]++};

if($i ~ /,Irmão/)
    {rela["Irmão"]++};

if($i ~ /,Tio Paterno/)
    {rela["Tio Patero"]++}

if($i ~ /,Tio Materno/)
    {rela["Tio Materno"]++};

if($i ~ /,Pai/)
    {rela["Pai"]++};

if($i ~ /,Pais/)
    {rela["Pais"]++};

if($i ~ /,Sobrinho Materno/)
    {rela["Sobrinho Materno"]++};

if($i ~ /,Sobrinho Paterno/)
    {rela["Sobrinho Paterno"]++};

if($i ~ /,Primo/)
    {rela["Primo"]++};

if($i ~ /,Avo Materno/)
    {rela["Avo Materno"]++};

if($i ~ /,Avo Paterno/)
    {rela["Avo Paterno"]++};

if($i ~ /,Tio Avo Materno/)
    {rela["Tio Avo Materno"]++};

if($i ~ /,Tio Avo Paterno/)
    {rela["Tio Avo Paterno"]++};

if($i ~ /,Neto Paterno/)
    {rela["Neto Paterno"]++};

if($i ~ /,Neto Materno/)
    {rela["Neto Materno"]++};

if($i ~ /,Sobrinho Bisneto Paterno/)
    {rela["Sobrinho Bisneto Paterno"]++};

if($i ~ /,Sobrinho Bisneto Materno/)
    {rela["Sobrinho Bisneto Materno"]++};
}

```

```
}  
  
END {  
  for (r in rela) {print "Da relação  " r "  há  " rela[r] "  relações"}  
}
```