Universidade do Minho
Escola de Engenharia

Trabalho Prático 2 - Conversor GPX-KML

Mestrado Integrado em Engenharia Informática

Processamento de Linguagens
2º Semestre
2017-2018

António Jorge Monteiro Chaves,A75870
Carlos José Gomes Campos,A74745
Luis Miguel Bravo Ferraz,A70824

Maio

Conteúdo

1 Introdução

Este trabalho tem como objetivos o aumento da experiência de uso do ambiente Linux e de o uso de algumas ferramentas que dão suporte à programação, como o gerador de filtros de texto FLEX. Através da utilização desta ferramenta, tivemos de por à prova os nossos conhecimentos sobre Expressões Regulares, uma vez que, tal como estava predefinido nos objetivos, através destas e com auxilio do gerador de filtros de texto, conseguimos extrair a informação de um tipo de ficheiros e ainda criar novos tipos ficheiros com essa informação. Foram propostos 5 enunciados, segundo as regras de atribuição destes a cada grupo, o nosso grupo ficou com o "Processamento de Trilhos GPS".

2 Processamento de Trilhos GPS

O problema que nos foi atribuido, passa por transformar um ficheiro, ou um conjunto de ficheiros que se encontram no formato ".gpx", para o formato ".kml", de modo a que estes possam ser consultados no respetivo visualizador.

2.1 Análise do texto-fonte

Depois de algumas pesquisas pela internet e da análise de alguns ficheiros no formato ".gpx", reparamos que existiam certas *tag's* que continham informação que não era necessária no formato ".kml"e que isso não impedia a representação dos trilhos nos respetivos visualizadores. De seguida, temos um pequeno exemplo, muito geral, de um ficheiro no formato ".gpx"que serve de ilustração e suporte para a estratégia que o grupo decidiu adoptar.

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<gpx>
  <trk>
    <trkseg>
      <trkpt lat="43.9548034" lon="4.8858287">
        <ele>74.411</ele>
        <time>2018-05-02T16:36:21Z</time>
        <hdop>32</hdop>
        <extensions>
          <speed>0.037</speed>
        </extensions>
      </trkpt>
      <trkpt lat="43.9548021" lon="4.8858284">
        <ele>74.382</ele>
        <time>2018-05-02T16:36:23Z</time>
        <hdop>32</hdop>
        <extensions>
          <speed>0.015</speed>
        </extensions>
      </trkpt>
    </trkseg>
  </trk>
</gpx>
```

2.2 Ações semânticas

Depois da análise do texto-fonte, achamos que apenas era necessário a extração da informação que se encontra dentro das *tag's* "<trkpt>"e "<ele>", filtrando apenas os respetivos valores.

2.3 Estruturas Globais de Dados

Uma vez que precisamos de gerar um ficheiro ".kml", tivemos a necessidade de criar um apontador do tipo *FILE*, de modo a conseguir a comunicação entre o programa e o ficheiro. Contudo, apenas precisamos de escrever os valores que estão nas *tag's* dos ficheiros ".gpx"nos ficheiros ".kml", não houve necessidade de criar mais estruturas de dados, apenas criamos várias variáveis do tipo *double* para guardar os valores das coordenadas. Contudo, reparamos que a ordem das coordenadas nos ficheiros era diferente, nos ficheiros ".gpx"aparecem "lat", "lon"e "ele"e nos ficheiros ".kml", aparecem "lon", "lat"e "ele", resolvemos isto apenas escrevendo os pontos no ficheiro depois de obtermos o valor da *tag* "<ele>".

```
%%
" lon\=\""{real}    {
                        lon=atof(yytext+6);
                        sumLon+=lon;
                    }
" lat\=\""{real}    {
                        lat=atof(yytext+6);
                        sumLat+=lat;
                    }
\<ele\>{real}       {
                        nPoints++;
                        ele=atof(yytext+5);
                        fprintf(file, "%f,%f,%f ",lon,lat,ele);
                    }
.                   {;}

%%
```

2.4 Filtro de Texto-Gerador Flex

Depois de analisada a informação, de analisados os respetivos padrões e o respetivo armazenamento, desenvolveu-se o filtro de texto utilizando o gerador FLEX.

- **Funções Auxiliares:** Achamos necessário recorrer a três funções auxiliares, a primeira foi a função "*header*", que nos permite abrir o ficheiro no formato ".gpx", para leitura, e guarda o seu nome numa variável auxiliar. Nesta variável auxiliar, a função substitui as posições que correspondem ao formato original, por o formato pretendido, neste caso o ".kml"e cria um novo ficheiro com o nome da respetiva variável. Por fim, escreve no novo ficheiro as *tag's* que pertencem ao formato pretendido, e os repetivos valores, tanto das coordenadas como o número do *Track*(trilho) analisado. Em seguida, criamos a função "*lookat*"de modo a calcular a média das coordenadas longitude e latitude que será necessária na *tag* "<Lookat>", e

escrevemos os respetivos valores e as repetivas *tag's* no ficheiro. Por fim, criamos a função "<footer>"que escreve no ficheiro o fecho das *tag's*, nos põe todos os valores das variáveis globais a zero e ainda fecha o ficheiro.

- **Definições:** Criámos uma definição, que é a de número real, porque precisamos apenas de trabalhar com números reais.
- **Regras:** A nossa estratégia passou por converter a *string* que obtivemos num *float* e guarda-la numa variável, apenas escrevendo no ficheiro depois de ser obtida a última coordenada(motivo falado anteriormente).
- **Main:** Por fim criamos a função main que nos permite converter mais do que um ficheiro ao mesmo tempo.

3 Implementação

```
%{
#include <string.h>
FILE* file;
double lon=0.0;
double lat=0.0;
double ele=0.0;
double sumLon=0.0;
double sumLat=0.0;
int nPoints=0;
%}

real [+-]?[0-9]+(\.[0-9]+)?

%%

" lon\=\""{real}    {
                        lon=atof(yytext+6);
                        sumLon+=lon;
                    }
" lat\=\""{real}    {
                        lat=atof(yytext+6);
                        sumLat+=lat;
                    }
\<ele\>{real}       {
                        nPoints++;
                        ele=atof(yytext+5);
                        fprintf(file, "%f,%f,%f ",lon,lat,ele);
                    }
.                   {;}
```

```
%%
void header(const char* nm, int i){
    yyin=fopen(nm,"r");
    char* name=strdup(nm);
    int l=strlen(name);
    name[l-3]='k';
    name[l-2]='m';
    name[l-1]='l';
    file=fopen(name,"w");
    fprintf(file,"<?xml version=\"1.0\"encoding=\"UTF-8\"?>\n<kml>\n\t<Document>
    \n\t\t<Placemark>\n\t\t\t<name>Track n°%d</name>\n\t\t\t<visibility>1</visibility>
    \n\t\t\t<open>1</open>\n\t\t\t<Styleid=\"red\">\n\t\t\t\t<LineStyle>\n\t\t\t\t\t
    <color>C81400FF</color>\n\t\t\t\t\t<width>4</width>\n\t\t\t\t</LineStyle>\n\t\t\t
    </Style>\n\t\t\t<LineString>\n\t\t\t\t<coordinates>\n",i);
}
void lookat(){
    fprintf(file,"\n\t\t\t\t</coordinates>\n\t\t\t\t<LookAt>\n\t\t\t\t\t
    <longitude>%f</longitude>\n\t\t\t\t\t<latitude>%f</latitude>\n\t\t\t\t\t
    <altitude>0</altitude>\n\t\t\t\t\t<heading>0</heading>\n\t\t\t\t\t<tilt>45
    </tilt>\n\t\t\t\t\t<altitudeMode>clampToGround</altitudeMode>\n\t\t\t\t
    </LookAt>",sumLon/nPoints,sumLat/nPoints);
}
void footer(){
    fprintf(file,"\n\t\t\t</LineString>\n\t\t</Placemark>\n\t</Document>\n</kml>");
    sumLon=0.0;
    sumLat=0.0;
    nPoints=0;
    fclose(file);
}
int main(int argc, char const *argv[]){
    if(argc==1){
        file=stdout;
        yylex();
    }else{
        for(int i=1;i<argc;i++){
            header(argv[i],i);
            yylex();
            lookat();
            footer();
        }
    }
    return 0;
}
```

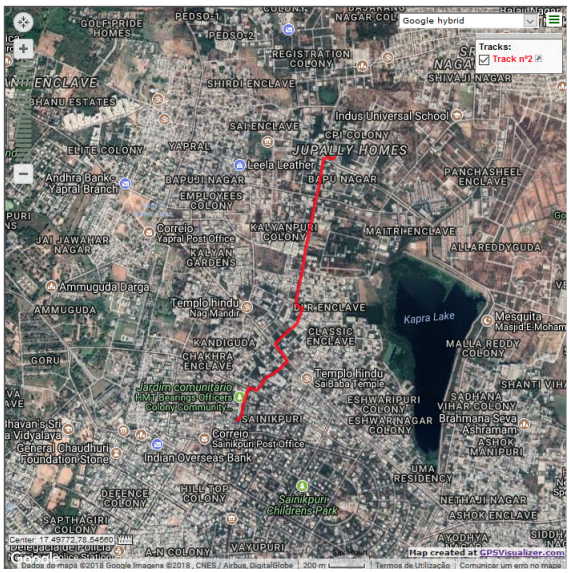
3.1 Ficheiro .gpx

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<gpx>
  <trk>
    <trkseg>
      <trkpt lat="43.9548034" lon="4.8858287">
        <ele>74.411</ele>
        <time>2018-05-02T16:36:21Z</time>
        <hdop>32</hdop>
        <extensions>
          <speed>0.037</speed>
        </extensions>
      </trkpt>
      <trkpt lat="43.9548021" lon="4.8858284">
        <ele>74.382</ele>
        <time>2018-05-02T16:36:23Z</time>
        <hdop>32</hdop>
        <extensions>
          <speed>0.015</speed>
        </extensions>
      </trkpt>
    </trkseg>
  </trk>
</gpx>
```

3.2 Output-Ficheiro .kml

```
<?xml version="1.0" encoding="UTF-8"?>
<kml>
    <Document>
        <Placemark>
            <name>Track n°1</name>
            <visibility>1</visibility>
            <open>1</open>
            <Style id="red">
                <LineStyle>
                    <color>C81400FF</color>
                    <width>4</width>
                </LineStyle>
            </Style>
            <LineString>
                <coordinates>
                    4.885829,43.954803,74.411000 4.885828,43.954802,74.382000
                </coordinates>
                <LookAt>
                    <longitude>4.885829</longitude>
                    <latitude>43.954803</latitude>
                    <altitude>0</altitude>
                    <heading>0</heading>
                    <tilt>45</tilt>
                    <altitudeMode>clampToGround</altitudeMode>
                </LookAt>
            </LineString>
        </Placemark>
    </Document>
</kml>
```

3.3 Output-Google Earth



4 Conclusão

Este trabalho prático foi nos proposto com o objetivo de aplicar os conhecimentos adquiridos nas aulas práticas relativamente à aplicação de filtros de texto em FLEX. A nossa principal dificuldade deveu se ao facto de as coordenadas alterarem a sua ordem de ficheiro para ficheiro, algo que só conseguimos ultrapassar depois de analisar cuidadosamente ambos os ficheiros. Concluindo, conseguimos fazer tudo que nos foi proposto no enunciado e ainda conseguimos aprofundar os conhecimentos tanto da utilização de filtros de texto, como do uso de expressões regulares.