

Universidade de Brasília

Departamento de Ciência da Computação



Lista de Exercícios 4

Organização de Arquivos

Autores:

Giovanni M Guidini 16/0122660

Vitor F Dullens 16/0148260

Lista de Exercícios 4

Alunos: Giovanni Guidini 16/0122660; Vitor Dullens 16/0148260.
Prof. Oscar Gaidos

Organização de Arquivos

CIC 116327

Data: 1 de Maio de 2018

Exercício 1

Partindo de uma imagem no format DICOM (qualquer imagem), desenvolver um programa em c ou c++ para extrair os metadados de dita imagem. Fazer isto para mais de uma imagem.

Código

O código para esta questão está no apêndice A. Para este código utilizamos uma biblioteca local própria que criamos para ajudar a organizar o código. Testamos em 4 imagens DICOM.

Os arquivos DICOM são arquivos binários, facilitando a obtenção da imagem, que é a parte mais importante, mas seus metadados estão estruturados de maneira inteligente, num sistema de **Data Elements**. Um **Data Element** é formado por uma tag, de 2 números, um tipo (VR), um tamanho, e o valor, nesta ordem, mas existem algumas variações.

A complexidade de ler os arquivos deste tipo está na grande quantidade de tags diferentes, sintaxes diferentes possíveis, etc. Para tanto precisamos de vários arquivos adicionais, com as tags e os VRs possíveis.

Nosso programa mostra os metadados na forma:

```
-----+
TAG: (gggg,eeee) - TagName
VR: VRAbbreviation
LEN: ElLen
VALUE
-----+
```

Onde gggg,eeee são os valores do Grupo e do Elemento, respectivamente. Juntos fazem a tag daquele elemento. TagName é o nome daquele elemento.

VRAbbreviation é a abreviação do tipo do valor daquele elemento.

ElLen é o tamanho em bytes que o valor do elemento ocupa.

VALUE é o valor daquele elemento, mostrado de acordo com seu tipo. **Saídas:**

Saídas

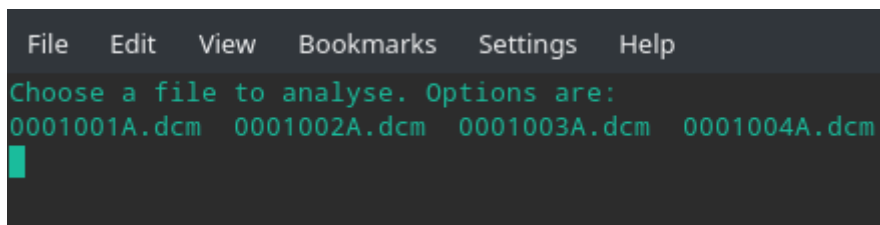


Figura 1: Menu de Escolha de arquivo

```

File chosen: 0001001A.dcm
Processed 2556 tags
TAG: (0x2,0) - FileMetaInformationGroupLength
VR: UL
LEN: 0x4
0xffffffffce
-----+
Forwarded 2 bytes because of specialVR
TAG: (0x2,0x1) - FileMetaInformationVersion
VR: OB
LEN: 0x2
0
-----+
TAG: (0x2,0x2) - MediaStorageSOPClassUID
VR: UI
LEN: 0x1c
1.2.840.10008.5.1.4.1.1.1.1
-----+
TAG: (0x2,0x3) - MediaStorageSOPInstanceUID
VR: UI
LEN: 0x36
1.3.51.0.7.757147185.51744.5191.40044.55318.30472.3805
-----+
TAG: (0x2,0x10) - TransferSyntaxUID
VR: UI
LEN: 0x14
1.2.840.10008.1.2.1
-----+
TAG: (0x2,0x12) - ImplementationClassUID
VR: UI
LEN: 0xc
1.3.51.0.1.3
-----+
TAG: (0x2,0x13) - ImplementationVersionName
VR: SH
LEN: 0xe
AGFA DTF1.0.95
-----+

```

Figura 2: Início dos metadados

```

+----- Still in the loop
TAG: (0x8,0x100) - CodeValue
VR: SH
LEN: 0x8
R-10214
+----- Still in the loop
TAG: (0x8,0x102) - CodingSchemeDesignator
VR: SH
LEN: 0x4
SNM3
+----- Still in the loop
TAG: (0x8,0x104) - CodeMeaning
VR: LO
LEN: 0x10
postero-anterior
+----- Still in the loop
Backwarded 2 bytes because of implicitVR
TAG: (0xfffe,0xe00d) - EndOfItem
VR: Im
LEN: 0
OUT OF THE LOOP: EndOfItem
+----- Still in the loop
Backwarded 2 bytes because of implicitVR
TAG: (0xfffe,0xe0dd) - EndOfSequence
VR: Im
LEN: 0
OUT OF THE LOOP: EndOfSequence
-----+
TAG: (0x2050,0x20) - PresentationLUTShape
VR: CS
LEN: 0x8
IDENTITY
-----+
Forwarded 2 bytes because of specialVR
TAG: (0x7fe0,0x10) - PixelInformation-TheImage
VR: OB
LEN: 0x657600
END OF METADATA
-----+
[gguidini@SilverAce LE4]$ █

```

Figura 3: Fim dos metadados

A Exercício 1

```
1 #include <bits/stdc++.h>
2 #include "HeaderElements.h"
3 #define lsDcm() std::system("ls *.dcm")
4 #ifdef WIN32
5     #define CLEAR "cls"
6 #else
7     #define CLEAR "clear"
8 #endif
9
10 using namespace std;
11
12 // function headers
13 HE::Element readNextMeta(ifstream* fd);
14 void readData(ifstream* fd, string vr, unsigned int len, HE::Element el);
15 // Some images have preamble followed by prefix DICM before metadata, others don't
16 // This functions will skip preamble if it exists.
17 void managePreamble(ifstream* fd){
18     fd->seekg(128);
19     char prefix[4];
20     fd->read(prefix, 4);
21     if(strcmp(prefix, "DICM") != 0){
22         printf("Este arquivo nao tem o preambulo!!!\n");
23         fd->seekg(0);
24     }
25 }
26 // A lot of problems were happening when comparing strings, so decided to do it
27 // the hard way
28 bool specialVR(char* vr){
29     if(vr[0] == 'O'){
30         if(vr[1] == 'B' || vr[1] == 'W' || vr[1] == 'F'){
31             return true;
32         }
33     }
34     else if(vr[0] == 'S' && vr[1] == 'Q'){
35         return true;
36     }
37     else if(vr[0] == 'U'){
38         if(vr[1] == 'T' || vr[1] == 'N'){
39             return true;
40         }
41     }
42     return false;
43 }
44
45 bool implicitVR(char* vr){
46     if(vr[0] > 90 || vr[0] < 65){
47         return true;
48     }
49     else if(vr[1] > 90 || vr[1] < 65){
50         return true;
51     }
52     return false;
53 }
54
55 void readData(ifstream* fd, string vr, unsigned int len, HE::Element el){
56     if(vr == "SQ" || len == (unsigned int) -1){
57         // Sequencia de itens sem valor definido.
```

```

57     printf("Sequence of Undefined Length\n");
HE::Element probe = HE::Element();
59     while(probe.getName() != "EndOfItem" && probe.getName() != "EndOfSequence
"){
        cout << "└────────────────── Still in the loop" << endl;
61         probe = readNextMeta(fd);
        }
63     return;
}
65 else if(el.getName() == "EndOfItem" || el.getName() == "EndOfSequence"){
    // Sequence delimiters
67     cout << "OUT OF THE LOOP: " << el.getName() << endl;
    return;
69 }
else if(el.getName() == "PixelInformation-TheImage"){
71     // Header delimiter
    cout << "END OF METADATA\n";
73     return;
}
75 else if(len == 0){
    cout << "LEN IS ZERO - NO DATA\n";
77 }
else if(el.compareVR("UL") || el.compareVR("OB") || el.compareVR("FL") ||
79     el.compareVR("US") || el.compareVR("SS")){
    // Numered Values
81     char buff[len+1];
    fd->read(buff, len);
83     if(el.compareVR("UL")){
        unsigned int out;
85         out = (buff[3] << 24) | (buff[2] << 16) | (buff[1] << 8) | buff[0];
        printf("%#x\n", out);
87     }
    else if(el.compareVR("OB")){
89         int8_t out = buff[0];
        printf("%d\n", out);
91     }
    else if(el.compareVR("FL")){
93         float out;
        out = (buff[3] << 24) | (buff[2] << 16) | (buff[1] << 8) | buff[0];
95         cout << out << endl;
    }
97     else if(el.compareVR("US")){
        unsigned short out;
99         out = (buff[1] << 8) | (0x00FF & buff[0]);
        cout << out << endl;
101    }
    else{
103        short out;
        out = (buff[1] << 8) | (0x00FF & buff[0]);
105        cout << out << endl;
    }
107 }
else if(el.compareVR("DA") || el.compareVR("TM") || el.compareVR("DT")){
109     // Date strings
    char buff[len+1];
    fd->read(buff, len);
111     buff[len] = '\0';
    if(el.compareVR("DA")){
113         // DA are arranged YYYYMMDD
115

```

```

117         char Y[5], M[3], D[3];
118         Y[0] = buff[0]; Y[1] = buff[1]; Y[2] = buff[2]; Y[3] = buff[3]; Y[4]
= '\0';
119         M[0] = buff[4]; M[1] = buff[5]; M[2] = '\0';
120         D[0] = buff[6]; D[1] = buff[7]; D[2] = '\0';
121         printf("%s/%s/%s\n", D,M,Y);
122     }
123     else if (el.compareVR("TM")){
124         // TM is HHMMSS.FFFFFFFF
125         printf("%c%c:%c%c:%c%c\n", buff[0], buff[1], buff[2], buff[3], buff
[4], buff[5]);
126     }
127     else{
128         // DT is YYYYMMDDHHMMSS.FFFFFFFF&ZZXX
129         char Y[5], M[3], D[3];
130         Y[0] = buff[0]; Y[1] = buff[1]; Y[2] = buff[2]; Y[3] = buff[3]; Y[4]
= '\0';
131         M[0] = buff[4]; M[1] = buff[5]; M[2] = '\0';
132         D[0] = buff[6]; D[1] = buff[7]; D[2] = '\0';
133         printf("%s/%s/%s ", D,M,Y);
134         printf("%c%c:%c%c:%c%c ", buff[8], buff[9], buff[10], buff[11], buff
[12], buff[13]);
135         if (buff[20] == '+' || buff[20] == '-'){
136             printf("UTF%c%c%c", buff[20], buff[21], buff[22]);
137         }
138         printf("\n");
139     }
140     else{
141         // String Values
142         char buff[len+1];
143         fd->read(buff, len);
144         buff[len] = '\0';
145         printf("%s\n", buff);
146     }
147 }

148 // tag for pixel data (7FE0,0010) ?
HE::Element readNextMeta(ifstream* fd){
149     char buff[128], vr[10];
150     uint16_t tag[2];
151     unsigned int len;
152     fd->read(buff, 4);
153     // DEBUG PRINT printf("buff for tag: %x %x %x %x\n", buff[0], buff[1], buff
[2], buff[3]);
154     tag[0] = (buff[1] << 8) | (0x00FF & buff[0]);
155     tag[1] = (buff[3] << 8) | (0x00FF & buff[2]);
156     fd->read(vr, 2);
157     // Dealing with cases OB, OW, OF, SQ, UT or UN for explicit VR, which have 2
reserved bytes
158     if (specialVR(vr)){
159         // forwards two bytes
160         cout << "Forwarded 2 bytes because of specialVR\n";
161         fd->read(buff, 2);
162         fd->read(buff, 4); // special values have 4 bytes of length
163         len = (buff[3] << 24) | (buff[2] << 16) | (buff[1] << 8) | buff[0];
164     }
165     // Some cases may contain implicit vr (?)
166     else if (implicitVR(vr)){
167         // backwards two bytes

```

```

171     cout << "Backwarded 2 bytes because of implicitVR\n";
172     int curr = fd->tellg();
173     fd->seekg(curr-2);
174     strcpy(vr, "Implicit\0");
175     fd->read(buff, 4); // special values have 4 bytes of length
176     len = (buff[3] << 24) | (buff[2] << 16) | (buff[1] << 8) | buff[0];
177 }
178 else{
179     fd->read(buff, 2);
180     len = (buff[1] << 8) | buff[0];
181 }
182 HE::Element specs = HE::getElement(make_pair(tag[0], tag[1]));
183 printf("TAG: (%#x,%#x) - %s\n", tag[0], tag[1], specs.getName().c_str());
184 printf("VR: %c%c\n", vr[0], vr[1]);
185 printf("LEN: %#x\n", len);
186 readData(fd, string(vr), len, specs);
187 return specs;
188 }
189 int main(){
190     string file;
191     system(CLEAR);
192     cout << "Choose a file to analyse. Options are:" << endl;
193     lsDcm();
194     cin >> file;
195     system(CLEAR);
196     cout << "File chosen: " << file << endl;
197     ifstream fd (file, ios::binary);
198     if(!fd){
199         cout << "Error 404 - File not Found\n";
200         return 1;
201     }
202     managePreamble(&fd);
203     HE::makeTag();
204     unsigned int g, e, i = 0;
205     HE::Element prev = HE::Element();
206     prev.getTag(&g, &e);
207     while(!(g == 0x7FE0 && e == 0x0010)){
208         prev = readNextMeta(&fd);
209         cout << "-----+" << endl;
210         prev.getTag(&g, &e);
211         // DEBUG PRINT cout << "Tellg:" << fd.tellg() << endl;
212         // DEBUG PRINT printf("PrevTag: %#x,%#x\n", g,e);
213         i++;
214     }
215     return 0;
216 }

```

Listing 1: "Code for Exercise 1"