Structuur van computer programma's II
Opgave project "fortunepp"

Dirk Vermeir
Departement Informatica
Faculteit Wetenschappen
Vrije Universiteit Brussel

November 1999

Inhoudsopgave

1 Inleiding: fortune **2**

2 Opgave: fortunepp **2**

2.1 Functionele eisen 2

2.2 Niet-functionele eisen 4

3 Ontwerp hints **5**

4 Test data **7**

5 Optionele uitbreidingen **7**

6 Vragen **8**

7 Indienen **8**

1 Inleiding: fortune

(Deze tekst is ook beschikbaar als een postscript bestand **stru2-1999.ps**. dat bvb. op weny kan afgedrukt worden).

Fortune is een unix/linux programma dat aangeroepen wordt dmv een bevel

```
fortune
```

waarop het programma een willekeurig citaat uit een gegeven verzameling van citaten op de de "standard output" zal schrijven.

Een vb. citaat:

> To err is human – but it feels divine.
> – *Mae West*

2 Opgave: fortunepp

Maak een programma "**fortunepp**" in C++ dat voldoet aan de onderstaande functionele[1] en niet-functionele[2] eisen ("requirements").

Een demo versie van **fortunepp** is beschikbaar.

2.1 Functionele eisen

Het programma kan op 2 manieren opgeroepen worden

1. **`fortunepp -i outputfile inputfile..`**

   Dit noemen we de "index" functie van het programma. In dit geval accepteert het programma dus minstens 2 (mogelijks meer) argumenten:

   - De **-i** optie (**`-i`** staat voor "indexeer").
   - De naam van een output file.
   - De namen van 0 of meer input files.

[1] Functionele eisen leggen vast **wat** het programma moet doen.

[2] Niet-functionele eisen leggen vast aan welke andere voorwaarden het programma moet voldoen, bvb. onder welk operating system het moet werken, welke programmeertalen of bibliotheken mogen gebruikt worden, etc.

Het programma zal dan een “index structuur” (zie sectie 3) opbouwen voor alle citaten die in de diverse input files voorkomen. Bovendien zal het programma die structuur wegschrijven op de output file.

De structuur van de input files is eenvoudig: het zijn tekst files met citaten waarbij citaten worden beëindigd door een lijn met in de eerste positie een percent (“%”) teken.

Hieronder een stukje van een mogelijke input file:

```
A banker is a fellow who lends you his umbrella when
the sun is shining and wants it back the minute it
begins to rain.
-- Mark Twain
%
A classic is something that everyone wants to have read
and nobody wants to read.
-- Mark Twain, "The Disappearance of Literature"
%
A horse!  A horse!  My kingdom for a horse!
-- Wm. Shakespeare, "Henry VI"
%
A hundred years from now it is very likely that [of Twain's
works] "The Jumping Frog" alone will be remembered.
-- Harry Thurston Peck (Editor of "The Bookman"), January 1901.
%
A is for Apple.
-- Hester Pryne
%
A kind of Batman of contemporary letters.
-- Philip Larkin on Anthony Burgess
%
A light wife doth make a heavy husband.
-- Wm. Shakespeare, "The Merchant of Venice"
%
```

De index structuur zal het makkelijk maken om, uitgaande van een woord, alle citaten terug te vinden waarin dat woord voorkomt.

2. **`fortunepp indexfile zinsnede..`**

   Dit noemen we de “query” versie van het programma. In dit geval accepteert het programma dus 1 of meer argumenten.

Het eerste argument is de naam van een bestand dat een index bevat zoals aangemaakt door de "index" versie van het programma.

De volgende argumenten zijn van een van de volgende vormen:

- een woord, bvb. **`divine`**, of
- een rij van woorden tussen dubbele haakjes, bvb. **`"mae west"`**. Zulk een argument wordt hierna een "zin" argument genoemd.

Het programma zal dan een **willekeurig** citaat teruggeven dat alle woorden bevat die voorkomen in een van de opgegeven zinsneden. Bovendien zal, voor elk zin argument het citaat de woorden in de zin bevatten, in de zelfde volgorde en zonder andere tussenliggende woorden.

Bvb. kan voor het bevel

```
fortunepp myindex divine "mae west"
```

(waarbij **`myindex`** een eerder aangemaakte index file is) het voorbeeld citaat uit sectie 1 worden getoond.

2.2 Niet-functionele eisen

1. Het programma moet volledig en uitsluitend in C++ geschreven worden.
2. Het programma moet correct werken op **wendy**.
3. Enkel de C++ (of C) standaard library mag gebruikt worden.
4. **Eigen implementaties van containers zijn niet toegestaan, enkel standaard library containers zoals map, set, vector, mogen gebruikt worden.**
5. Het gebruik van expliciete pointers moet zoveel mogelijk vermeden worden.
6. De query versie van het programma moet zoveel mogelijk gebruik maken van de informatie in de index file die door de "index" versie wordt aangemaakt.
7. Het programma mag geen "ingebouwde" file namen of citaten bevatten.
8. De source (C++) bestanden dienen de suffix ".cc", ".C" of ".cpp" te gebruiken. De "header files" met declaraties dienen de suffix ".h" te gebruiken.

9. Alle source bestanden alsook een “Makefile” dienen zich in een directory “structuur2/project99” te bevinden. De volgende reeks van bevelen zal resulteren in een correct uitvoerbaar programma “fortunepp” in dit directory:

```
make clean
rm -f fortunepp *.o
make fortunepp
```

10. U kunt zelf een geschikte definitie van “woord” bepalen. Deze definitie mag niet minder gesofistikeerd zijn dan de volgende.

    > Een woord bestaat enkel uit alphabetische tekens (d.w.z. letters). Om een woord in een tekst te herkennen volstaat het te zoeken naar het eerste alphabetische teken en daar alle volgende tekens aan toe te voegen tot er een niet-alphabetisch teken wordt gevonden of tot het einde van de tekst is bereikt.

11. Voor de indexering en zoek operaties zullen alle woorden naar kleine letters worden geconverteerd. B.v.b. zal

    `fortunepp myindex DIVINE "Mae West"`

    precies hetzelfde effect hebben als

    `fortunepp myindex divine "mae west"`

3 Ontwerp hints

1. Een file in unix wordt beschouwd als een rij van bytes. Door middel van functies zoals

   `ifstream::seekg(unsigned long)`

   kan een open “file stream” (**`fstream`**) op een willekeurige positie gepositioneerd worden.
   Omgekeerd kan je de “huidige positie” in een open **ifstream** opvragen d.m.v. de functie

   `unsigned long ifstream::tellg()`

   Een bepaald aantal bytes inlezen kan door de functie

   `ifstream::read(char* buffer,unsigned long size)`.

   Het volgende programma leest bytes 100..299 uit file “input.txt”.

```
#include        <fstream>
int
main(int argc,char*argv[]) {
const char*      ifnm = "input.txt";

ifstream         ifile(ifnm);

if (!ifile) {
        cerr << "Error: cannot open " << ifnm << endl;
        exit(1);
        }

const int        START = 100;
const int        BUFSIZE = 200;
char             buf[BUFSIZE];

ifile.seekg(START);

if (!ifile) {
        cerr << "Error: cannot seek to "
             << START << endl;
        exit(2);
        }

ifile.read(buf,BUFSIZE);

if (!ifile) {
        cerr << "Error: cannot read "
             << BUFSIZE  << " bytes" << endl;
        exit(3);
        }

}
```

Meer informaties over filestreams in [Stroustrup97].

2. Een citaat kan geidentificeerd worden door
   - De (volledige) padnaam van de file waarin het zich bevindt.
   - De start positie in de file waarin het zich bevindt.
   - De lengte (in characters) van het citaat.

3. De geschatte grootte van het programma is circa 500 lijnen (of minder), inclusief header files.

4. Door de werking van de shell (zie **"Unix for beginners"**) zal, in de query versie, elke zinsnede een apart argument van het programma vormen, waarbij de aanhalingstekens (**"**) door de shell verwijderd werden.

   Bvb. zal bij het bevel

   ```
   fortunepp myindex divine "Mae West"
   ```

   gelden dat (de namen zijn zoals in de voorbeelden gezien in de cursus):

   - **argv[0]="fortunepp"**
   - **argv[1]="myindex"**
   - **argv[2]="divine"**
   - **argv[3]="Mae West"**

4 Test data

Een aantal files met (samen zo'n 12000) citaten bevinden zich in het wendy directory

```
/export/home3/staff/dvermeir/citaten
```

Men kan deze files gebruiken om te testen.
Een goede test is bvb

```
fortunepp -i myindex /export/home3/staff/dvermeir/citaten/*
```

die alle vb files in **/export/home3/staff/dvermeir/citaten** zal indexeren. Dit kan dan gevolgd worden door

```
fortunepp myindex divine "mae west"
```

5 Optionele uitbreidingen

De volgende uitbreidingen zijn optioneel; uiteraard zal het goed implementeren van een uitbreiding passend geapprecieerd worden. Je kan ook een eigen uitbreiding voorstellen.

1. Voorzie een "**`-a`**" optie die, in plaats van een willekeurig citaat te tonen dat aan de voorwaarden voldoet, **alle** citaten toont die aan de voorwaarden voldoen.

2. Maak een "cgi" programma van "fortunepp". Hiervoor mag je evt. een perl of bash script gebruiken dat zelf weer het "fortunepp" programma aanroept.

6 Vragen

Zoals steeds kan je vragen sturen naar **`structuur2@tinf.vub.ac.be`**.

7 Indienen

De uiterste indiendatum (de source code en Makefile wordt automatisch opgehaald uit het directory **`$HOME/structuur2/project99`**, zie punt 9 in sectie 2.2) is 25 februari 2000, 23:59.