資料工程 HW0

404410030 資工三 鄭光宇

系統需求

要執行這支程式，系統必須具備：

- Apache2.0、PHP5.0 以上
- 支援 C11 的 gcc
- make 指令

如何編譯

在專案目錄下輸入指令：

```
make
```

如何執行

1. 在專案同目錄下新增 `data` 目錄
2. 將 ettoday0.rec ~ ettoday5.rec 放在專案同目錄下的 `data` 目錄
3. 執行 `make` 編譯程式，編譯出 `parser` 程式
4. 執行 `./parser` 程式，程式輸出斷句、排序後的結果到 `dataset.txt`
5. 將 `dataset.txt` 和專題目錄下的 `index.php` 放到你的個人網頁資料夾 or `/var/www/html`
6. 在瀏覽器網址輸入 `http://localhost/你的個人網站/index.php` or `http://localhost/index.php` 進入網頁
7. 搜尋句子

實作部分

前置處理

依照作業規定，使用「。？！」等字元進行斷句。僅採用中文字開頭的句子，並且句子中的中文字必須有五個中文字以上（不包含五）。中文字的部分 Uncode 編碼我採用 `0x4E00` ~ `0x9FFF` 這個範圍。之後定義一個 `struct`，方便後面處理資料。

```c
1 parser.c
 1 #include <stdio.h>
 2 #include <stdlib.h>
 3 #include <ctype.h>
 4 #include <string.h>
 5 #include <wchar.h>
 6 #include <locale.h>
 7 #include <unistd.h>
 8 #include "postprocess.h"
 9
10 const char *file_name_prefix = "./data/ettoday";
11 const char *temp_out_file = "./sentences.txt";
12 const char *final_out_file = "./dataset.txt";
13 const char *file_name_postfix = ".rec";
14 const size_t file_numbers = 6;
15 const size_t file_prefix_length = 14;
16 const size_t buffer_limit = 8192;
17 const size_t chinese_min_len = 6; // 一句話要有5個字以上（不包含）
18
19 // 中文 Unicode 區
20 const unsigned long min_chinese = 0x4E00;
21 const unsigned long max_chinese = 0x9FFF;
22 // End 中文 unicode 區
23
24 const wchar_t *tokens = L"。？！\r\b\t\n?!"; // 斷句 tokens
25
26 typedef struct {
27     wchar_t *url; // 網頁 URL
28     wchar_t *title; // 新聞標題
29     wchar_t *context; // 新聞內文
30 } news_record;
```

定義判斷中文字範圍的工具函式。

```c
31
32 int8_t is_chinese(wchar_t c) {
33     unsigned long c_ = (unsigned long)c;
34     return ((c_>=min_chinese) && (c_<=max_chinese))?1:0;
35 }
36
```

斷詞部分。實作一個 dynamic array 去儲存斷好的句子。

```
43 int tokenize(wchar_t ***results, wchar_t *str) {
44     wchar_t **sentences=NULL, **s_ptr_t=NULL;
45     wchar_t *ptr=NULL, *buff=NULL;
46     int cnt = 0;
47     int cap = 32; // 初始容量，三十二個寬字元指標
48     sentences = (wchar_t**) malloc(sizeof(wchar_t*)*cap);
49     if (sentences==NULL) exit(1);
50     ptr = wcstok(str, tokens, &buff);
51     while(ptr!=NULL) {
52         unsigned long ch_cnt = 0;
53         wchar_t *ch_test_ptr = ptr;
54         if (ch_test_ptr!=NULL && is_chinese(ch_test_ptr[0])) { // 第一個字必須是中文字
55             while(ch_test_ptr!=NULL && *ch_test_ptr!=0) {
56                 ch_cnt += is_chinese(*ch_test_ptr);
57                 ++ch_test_ptr;
58             }
59         }
60         if (ch_cnt>=chinese_min_len) { // 一句中文必須達到6個字(含)以上
61             if(cnt==cap) { // buffer 滿了
62                 cap *= 2; // 增加一倍容量
63                 s_ptr_t = NULL;
64                 s_ptr_t = (wchar_t**) malloc(sizeof(wchar_t*)*cap);
65                 if (s_ptr_t==NULL) exit(2);
66                 memcpy(s_ptr_t, sentences, sizeof(wchar_t*)*cnt);
67                 free(sentences);
68                 sentences = s_ptr_t;
69                 s_ptr_t = NULL;
70             }
71             sentences[cnt] = NULL;
72             sentences[cnt] = (wchar_t*)malloc(sizeof(wchar_t)*(wcslen(ptr)+1));
73             if (sentences[cnt]==NULL) exit(4);
74             wcscpy(sentences[cnt], ptr);
75             ++cnt;
76         }
77         ptr = wcstok(NULL, tokens, &buff);
78     }
```

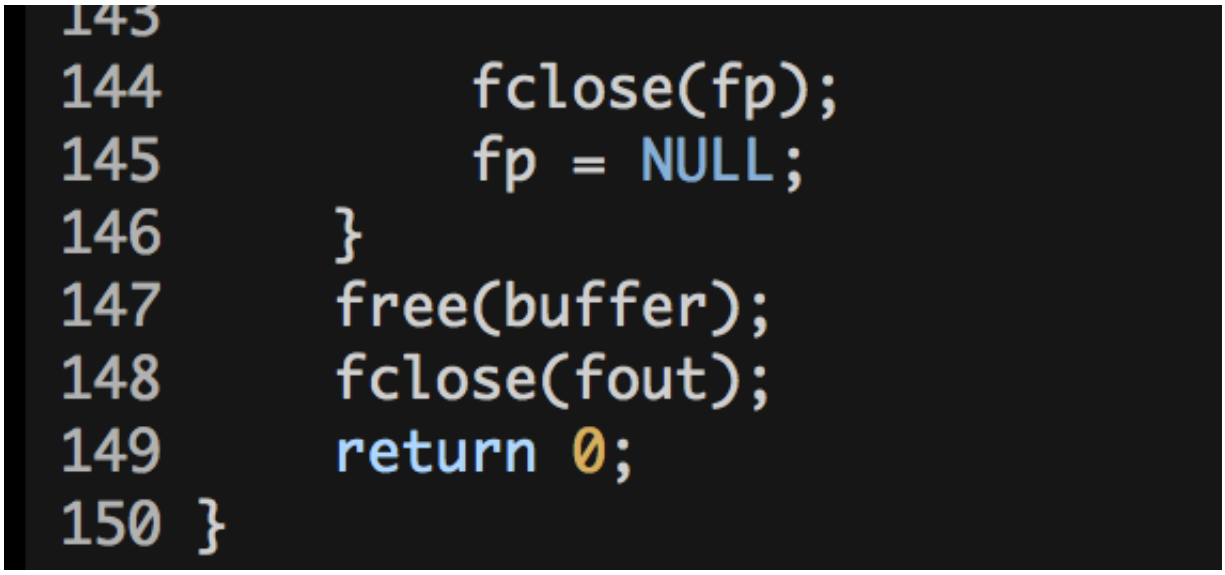
```
78     }
79     s_ptr_t = NULL;
80     s_ptr_t = (wchar_t**) malloc(sizeof(wchar_t*)*cnt);
81     if (s_ptr_t==NULL) exit(3);
82     memcpy(s_ptr_t, sentences, sizeof(wchar_t*)*cnt);
83     free(sentences);
84     sentences = s_ptr_t;
85     s_ptr_t = NULL;
86     *results = sentences;
87     return cnt;
88 }
```

工具函式，做一些空格、tab 等特殊字元的處理。

```
 90 void format_line(wchar_t *str) { // 假設輸入 str 只有一行內容
 91     wchar_t *ptr = str;
 92     while(ptr!=NULL && *ptr!=0) {
 93         if (*ptr==L'\t' || *ptr==L'\b' || *ptr==L'\r') *ptr=L' '; // 清除空格與其他東西
 94         if (*ptr==L'\n') {  // 清除結尾換行符號
 95             *ptr=0;
 96             break;
 97         }
 98         ++ptr;
 99     }
100 }
```

parser 部分。先找到 `@GAISRec` 行，之後連續讀取四行，得到標題、URL、內文等資料。對每個文章內文做完斷句後，再將該句子、來源標題與 URL，以 tab 分隔模式寫在同一行。

```
102 int parse(void) {
103     char filename[32];
104     wchar_t *buffer = NULL, *ptr=NULL;
105     wchar_t **sentences = NULL;
106     FILE *fp = NULL;
107     FILE *fout = NULL;
108     news_record one_record;
109     fout = fopen(temp_out_file, "wb");
110     if (fout==NULL) exit(9);
111     buffer = (wchar_t*)malloc(sizeof(wchar_t)*(buffer_limit+8));
112     if (buffer==NULL) exit(10);
113     for (size_t fno=0; fno<file_numbers; ++fno) {
114         get_file_path(filename, fno);
115         fp = fopen(filename, "rb");
116         if (fp==NULL) return -1;
117 
118         while(fgetws(buffer, buffer_limit, fp)!=NULL) {
119             if(wcsncmp(buffer, L"@GAISRec:", 9)!=0) continue; // 找下一筆資料的開頭
120             fgetws(buffer, buffer_limit, fp); // url
121             format_line(buffer);
122             one_record.url = (wchar_t*)malloc(sizeof(wchar_t)*(wcslen(buffer)+1));
123             wcscpy(one_record.url, buffer+3);
124             fgetws(buffer, buffer_limit, fp); // title
125             format_line(buffer);
126             one_record.title = (wchar_t*)malloc(sizeof(wchar_t)*(wcslen(buffer)+1));
127             wcscpy(one_record.title, buffer+3);
128             fgetws(buffer, buffer_limit, fp); // null
129             fgetws(buffer, buffer_limit, fp); // context
130             one_record.context = (wchar_t*)malloc(sizeof(wchar_t)*(wcslen(buffer)+1));
131             wcscpy(one_record.context, buffer);
132             int s_cnt = 0;
133             s_cnt = tokenize(&sentences, one_record.context); // 這裡做斷句
134             free(one_record.context); one_record.context=NULL;
135             for (size_t i=0; i<s_cnt; ++i) {
136                 fwprintf(fout, L"%ls\t%ls\t%ls\n", sentences[i], one_record.title, one_record.url); // 寫入句子到檔案
137                 free(sentences[i]); sentences[i]=NULL;
138             }
139             free(sentences); sentences=NULL;
140             free(one_record.url); one_record.url=NULL;
141             free(one_record.title); one_record.title=NULL;
142         }
```



取得所有斷句後，照句子字典序對句子排序。最後寫入 `dataset.txt` 檔案。

```
151
152 void read_and_sort(void) {
153     FILE *fp = NULL, *fout=NULL;
154     wchar_t **sentences = NULL, *buffer=NULL, **new_s_p=NULL, *new_row=NULL;
155     size_t cap = 1024;
156     fp = fopen(temp_out_file, "rb");
157     fout = fopen(final_out_file, "wb");
158     sentences = (wchar_t**) malloc(sizeof(wchar_t*)*cap);
159     if (sentences==NULL) exit(5);
160     buffer = (wchar_t*) malloc(sizeof(wchar_t)*(buffer_limit+8));
161     if (buffer==NULL) exit(6);
162
163     size_t cnt = 0;
164     while(fgetws(buffer, buffer_limit, fp)!=NULL) {
165         if (cnt==cap) {
166             cap*=2;
167             new_s_p = NULL;
168             new_s_p = (wchar_t**)malloc(sizeof(wchar_t*)*cap);
169             if (new_s_p==NULL) exit(7);
170             memcpy(new_s_p, sentences, sizeof(wchar_t*)*cnt);
171             free(sentences);
172             sentences = new_s_p;
173             new_s_p = NULL;
174         }
175         size_t len = wcslen(buffer);
176         new_row = NULL;
177         new_row = (wchar_t*)malloc(sizeof(wchar_t)*(len+1));
178         if (new_row==NULL) exit(8);
179         wcscpy(new_row, buffer);
180         sentences[cnt] = new_row;
181         new_row = NULL;
182         ++cnt;
183     }
184     fclose(fp); fp=NULL;
185     free(buffer); buffer=NULL;
186
187     postprocess(sentences, cnt); // sort in lexical order
```

```
188
189     for (size_t i=0; i<cnt; ++i) { // traverse sorted sentences
190         fputws(sentences[i], fout);
191         free(sentences[i]);
192         sentences[i] = NULL;
193     }
194     free(sentences);
195     fclose(fout); fout=NULL;
196 }
197
```

使用 `wcscmp` 比較句子 key 值大小。

```
1 postprocess.c
  1 #include "postprocess.h"
  2
  3 int wchar_cmp_func(const void *a, const void *b) {
  4     wchar_t **c = (wchar_t**)a;
  5     wchar_t **d = (wchar_t**)b;
  6     return wcscmp(*c, *d);
  7 }
  8
  9 void postprocess(wchar_t **str, size_t cnt) {
 10     qsort((void*)str, cnt, sizeof(wchar_t*), wchar_cmp_func);
 11 }
 12
```

主函式部分。

```
198 int main(void) {
199     setlocale(LC_ALL, ""); // 使用這個， fgetws 才不會出錯
200     parse();
201     read_and_sort();
202     return 0;
203 }
```

網頁介面部分

使用表單 + PHP，讓使用者查詢指定開頭的句子。後台呼叫 `grep` 找出 `dataset.txt` 中，符合的句子。並將結果寫入暫存檔。若使用者在同一個 session 並且查詢的內容是相同的，就直接讀取暫存檔。

```
 7     <body>
 8         <form action="?" method="get">
 9             <div class="form-group", align="left">
10                 <label for="search">搜尋：</label>
11                 <input type="search" class="form-control" id="search" name="search">
12             </div>
13             <button type="submit" class="btn btn-default">Go!</button>
14         </form>
```

```php
<?php
header("content-type:text/html;charset=utf-8");
session_start();
$page_max_row = 50;
$search_pattern = "—";
if (isset($_GET["search"])) {
    $search_pattern = stripslashes($_GET["search"]);
}
$page = 0;
if (isset($_GET["page"]) && $_GET["page"]>=0) {
    $page = $_GET["page"];
}
$page = preg_replace('/[^0-9]/', '', stripslashes($page));
$start_row = $page * $page_max_row;
$end_row = ($page+1) * $page_max_row;
if (isset($_SESSION['search']) && $_SESSION['search']===$_GET['search'] && isset($_SESSION['tmp_result']) && file_exists($_SESSION['tmp_result'])) {
    $tmp_result = $_SESSION['tmp_result'];
    // echo $tmp_result;
} else {
    $tmp_result = tempnam("/tmp", "search_sentence_");
    $command = "grep \"^" . $search_pattern . "\" ./dataset.txt > " . $tmp_result;
    exec($command, $outputs, $return_status);
    $_SESSION['tmp_result'] = $tmp_result;
    $_SESSION['search'] = $search_pattern;
}
```

搜尋：

咖啡

Go!

來源	內文
北京三聯韜奮書店24小時營業　靈感來自誠品	咖啡、書香上下樓相連，為讀者打造更佳閱讀體驗
萬聖節限定限量！床單幽靈造型甜點超吸睛	咖啡、甜食的價格都平易近人的魚缸咖啡，卻擁有著紮實的內涵，在生意已經步入軌道的情況下，仍然願意嘗試著創新與突破，這就是更難能可貴的地方了
龜鹿存骨本　老年趴趴走	咖啡、飲酒，都會促進了尿鈣的排泄，也影響鈣吸收、妨礙骨骼生成，所以現代人骨頭有問題的年齡層當然逐漸下降，年紀輕輕就腰酸背痛、脊椎側彎、駝背、膝蓋無力、環節不適，一大堆的老人症頭
「極光之愛」裡的浪漫花店　不限用餐時間的文創咖啡廳	咖啡不苦澀，奶泡很綿密，若真要說缺點是咖啡不夠燙口，反應給主廚是說啦花太燙口，不好拉，所以用這樣的溫度來製作
早安健康／一天__杯咖啡、30分鐘健走　消除脂肪肝	咖啡中含有大量屬於咖啡多酚的「綠原酸」（咖啡鞣酸），綠原酸有改善消化器官機能的作用，不止護肝還能減少體脂肪
抵銷酒精副作用！　一天一杯咖啡降低罹患肝癌風險	咖啡中含有綠原酸、咖啡因、多環胺類，都是很強的抗氧化劑，可降低致癌機率
夜貓子蠢蠢欲動中！網友最愛台北10大深夜咖啡館	咖啡也一樣以高溫烘焙的過程，必須如鑄劍般精密且專注，故特別以此命名
免出國！桃園就有棉花糖鬆餅　期間限定「老咩醉莓粒」	咖啡也很順口，剛好可以搭著鬆餅享用
桃園也有超吸睛的棉花糖鬆餅	咖啡也很順口，剛好可以搭著鬆餅享用

根據使用者所在 page，印出相應的 50 行結果。每一行都附帶原標題與連結在左側，而相關內文在右側。

像綠洲般存在　商業大樓附近的工業風咖啡館	咖啡散策：50＋風格咖啡館 絕少樑柱的寬敞室內，赭紅的清水磚牆，刻意裸露的金屬管線，棧木板與白鐵水管手工焊接成的桌椅層架，作舊的仿古燈具散發出溫暖的黃光……，一樣樣的元素，組成這個復古又摩登的工業風空間
一杯咖啡救了一條寶貴人命　公平貿易豆助女孩重新站起	咖啡散策：50＋風格咖啡館 那是一處，要用心、刻意才找得到的咖啡香，懷著探索的心情，越過台北與新北的福和橋，尋一段與咖啡獨處的時光
公平貿易外也支持台灣小農　追尋熱愛的早午餐咖啡館	咖啡散策：50＋風格咖啡館》 「hi, 日楞」，音譯自蒙古文的「海日楞」，意指「去愛」，是一種對生活的積極實踐—投注心力，去追尋生命中熱愛的人事物
感受跳的品味　海尼根新品Light選定台灣全球首賣	咖啡文化達人林東源昨天也驚喜現身，但他不賣咖啡，而是來分享咖啡館文化的多元角色
高雄千葉火鍋　咖啡放20分後變「豬血糕」	咖啡明明是液態，怎麼會變結成凍狀

1 2 3

最後在網頁尾端印出到每個 page 的按鈕，提供使用者選擇。

```php
            $handle = fopen($tmp_result, "r");

            echo "<div> <table class=\"table table-striped\">";
            echo "<tr> <th> 來源 </th> <th> 內文 </th> </tr>";
            $cnt = 0;
            while (($line = fgets($handle)) !== false) {
                if ($cnt >= $start_row && $cnt < $end_row) {
                    $arr = explode("\t", $line);
                    echo "<tr>";
                    echo "<td><a href=" . $arr[2] . ">" . $arr[1] . "</a></td>";
                    echo "<td>" . $arr[0] . "</td>";
                    echo "</tr>";
                }
                ++$cnt;
            }
            echo "</table></div>";
            // now $cnt has total line number
            $tot_pages = $cnt / $page_max_row; // floor
            $url = (isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] === 'on' ? "https" : "http") . "://$_SERVER[HTTP_HOST]" . explode('?', $_SERVER['REQUEST_URI'], 2 )[0];
            echo "<div align=\"center\">";
            for ($i=0; $i<$tot_pages; ++$i) {
                $newurl = $url . "?search=" . $search_pattern . "&page=" . $i;
                echo "<a href=\"" . $newurl . "\"><button type=\"button\" class=\"btn btn-default\">" . ($i+1) . "</button></a>";
            }
            echo "</div>";
            ?>
```

計算前置處理花費時間

計算前置處理花費時間：

```
[cky104u@csie0[9:42pm]~/Data-Engineering/HW0>time ./parser
15.503u 1.518s 0:18.81 90.4%    10+168k 0+0io 0pf+0w
```

從讀擋到排序輸出大約 18 秒左右。

GitHub

https://github.com/peter0749/Data-Engineering/tree/master/HW0