Національний технічний університет України

«Київський Політехнічний Інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота №5**

**З предмету «Логічне програмування»**

Виконав:

Студент  
IV курсу ФІОТ  
групи ІО-12  
Бута С. О.

Залікова книжка №1205

Київ-2015

**Завдання**

Розробка предиката для аналізу текстового файлу і створює список з термів слів і кількості їх повторень.

**Звіт**

Лістинг:

File.txt

qqq

wew fdf sdf sddd

ddd

ddd

dddd

ddddd dfdffdf qqq

qqqq

qqq ddd 222

22v 33333

33333 sedrsd sdsd d 222 qqq ddd 222

Пример выполнения

?- reconsult('l5.pro')

yes

?- init.

yes

?- gg(X).

X = [w(d, 1), w(sdsd, 1), w(sedrsd, 1), w('33333', 2), w('22v', 1), w('222', 3), w(qqqq, 1), w(dfdffdf, 1), w(ddddd, 1), w(d

dd, 1), w(ddd, 4), w(sddd, 1), w(sdf, 1), w(fdf, 1), w(wew, 1), w(qqq, 4)]

#include <amzi.h>

#include <windows.h>

#include <math.h>

#include <conio.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

BOOL APIENTRY DLLMain(HANDLE hModule,DWORD ul\_reason\_for\_call,LPVOID lpReserned)

{

return TRUE;

}

TF EXPFUNC GetL(ENGid eid)

{

RC rc;

int cnt=65;

TERM el, elem,head;

int i = 0;

int j = 0;

int k = 0;

int tmp = 0;

int flag = 0;

char\* w;

struct words

{

char word[256];

int count;

};

struct words arr\_words[500];

int count\_words = 0;

char buf[256];

char word[256];

FILE \*in;

for(i = 0; i<500; i++)

{

arr\_words[i].count = 0;

}

if ((in = fopen("file.txt", "r")) == NULL)

{

fprintf(stderr, "Cannot open input file.\n");

return 1;

}

count\_words = 0;

while (!feof(in))

{

fgets(buf, 256 , in);

i = 0;

j = 0;

while(buf[i] != '\0')

{

if(buf[i] != ' ')

{

word[j] = buf[i];

j++;

}

else

{

word[j] = '\0';

j = 0;

if(count\_words == 0)

{

strcpy(arr\_words[count\_words].word, word);

arr\_words[count\_words].count++;

count\_words++;

}

else

{

tmp = count\_words;

flag = 0;

for(k = 0; k < tmp; k++)

{

if(!strcmp(arr\_words[k].word, word))

{

arr\_words[k].count++;

flag = 1;

}

}

if(flag == 0)

{

strcpy(arr\_words[count\_words].word, word);

arr\_words[count\_words].count++;

count\_words++;

}

}

}

i++;

}

if(word[j-1] == '\n')

j--;

word[j] = '\0';

j = 0;

if(count\_words == 0)

{

strcpy(arr\_words[count\_words].word, word);

arr\_words[count\_words].count++;

count\_words++;

}

else

{

tmp = count\_words;

flag = 0;

for(k = 0; k < tmp; k++)

{

if(!strcmp(arr\_words[k].word, word))

{

arr\_words[k].count++;

flag = 1;

}

}

if(flag == 0)

{

strcpy(arr\_words[count\_words].word, word);

arr\_words[count\_words].count++;

count\_words++;

}

}

}

fclose(in);

lsMakeList(eid, &head);

for(i = 0; i<count\_words; i++)

{

lsMakeFA(eid, &elem, "w", 2);

w = arr\_words[i].word;

lsUnifyArg(eid, &elem, 1, cATOM, w);

lsMakeInt(eid, &el, arr\_words[i].count);

lsUnifyArg(eid, &elem, 2, cTERM, &el);

lsPushList(eid, &head, elem);

}

lsUnifyParm(eid, 1, cTERM, &head);

return TRUE;

}

PRED\_INIT aray\_preds[]=

{

{"gg", 1, GetL},

{NULL,0,NULL}

};

\_\_declspec(dllexport) RC EXPFUNC InitPreds(ENGid eid,void\* p)

{

RC rc =lsInitPreds(eid,aray\_preds);

return 0;

}