

Rješenja programskih zadataka iz prijemnog ispita

Utorak, 28.2.2012

7. [5] Napisati program koji broji koliko je puta pokrenut s različitom IP adresom. IP adresa se zadaje kao prvi i jedini argument programa. Primjer pokretanja:

```
./pokretbroj 195.29.149.197 # ispiše 1
./pokretbroj 127.0.0.1      # ispiše 2
./pokretbroj 195.29.149.197 # ispiše 2 (adresa 195.29.149.197 je već bila)
```

Rješenje:

```
#include <stdio.h>
#include <string.h>

int main(int argn, char *argv[]) {
    char ipaddr[24], nasao = 0;
    FILE *ipf, *cf;
    int count;

    cf = fopen("count.txt", "r+t");
    if (!cf) {
        cf = fopen("count.txt", "wt");
        count = 0;
    } else {
        fscanf(cf, "%d", &count);
    }

    ipf = fopen("ips.txt", "r+t");
    if (!ipf) {
        ipf = fopen("ips.txt", "wt");
    } else {
        while (fgets(ipaddr, sizeof(ipaddr), ipf)) {
            ipaddr[strlen(ipaddr)-1] = '\0'; // miče \n
            if (!strcmp(ipaddr, argv[1])) {
                nasao = 1;
                break;
            }
        }
    }

    if (!nasao) {
        fprintf(ipf, "%s\n", argv[1]);
        fseek(cf, 0, SEEK_SET);
        fprintf(cf, "%d", ++count);
    }

    printf("%d\n", count);

    fclose(ipf);
    fclose(cf);

    return 0;
}
```

8. [3.5] URL kodiranje zamjenjuje svaki nepoželjan znak znakom postotka iza kojeg slijedi dvoznamenkasta heksadekadska ASCII vrijednost nepoželjnog znaka. Primjerice "php%20je%20super%20100%25" je URL kodiran tekst "php je super 100%" (20_{16} je 32_{10} , odnosno ASCII vrijednost znaka ' ', a 25_{16} je ASCII vrijednost znaka '%').

Napišite funkciju za dekodiranje URL-kodiranog tekstualnog sadržaja zadanog argumentom funkcije u kojeg se sprema i rezultat. Prototip funkcije izgleda ovako:

```
void url_dekodiraj(char *tekst);
```

Napomena: Pretpostavite da je tekst uvijek moguće bez greške dekodirati. Nije potrebno znak '+' zamjenjivati razmakom.

Rješenje:

```
void url_dekodiraj(char *text) {
    int i, chr;
    for (; *text; ++text) {
        if (*text == '%') {
            sscanf(text, "%02X", &chr);
            *text = (char)chr;
            memmove(text+1, text+3, strlen(text+2));
        }
    }
}
```

9. [3.5] Napišite program za pretvaranje obojanih slika u crno-bijele ili sepia (ovisno o tome što je zadao korisnik). Za učitavanje i uređivanje slika koristite funkcije iz zaglavlja "picture.h" prikazanog na kraju zadatka. Primjer pokretanja:

```
./konvert slika1.png sepia slika2.jpeg
```

Zaglavlje "picture.h":

```
#ifndef _PICTURE_H_
#define _PICTURE_H_

typedef struct { /* sadržaj ove strukture je nebitan programeru */
} PIC; // tip PIC (opisnik slike)

typedef struct {
    unsigned char red, green, blue;
} PIX; // tip PIX (opisnik piksela)

// učitavanje slike iz datoteke; vraća NULL ako datoteka ne postoji ili
// nije slika
PIC* picLoad(const char *datoteka);
// stvaranje nove slike; parametri su širina i visina nove slike
PIC* picNew(int width, int height);

// spremanje slike u datoteku; vraća -1 ako je došlo do greške prilikom
// spremanja; tip slike automatski određuje prema ekstenziji
int picSave(PIC *pic, const char *datoteka);

void picFree(PIC *pic); // oslobađa memoriju koju slika zauzima

int picGetW(PIC *pic); // dohvaćanje dimenzija slike
int picGetH(PIC *pic);
```

```

// dohvaćanje/postavljanje piksela slike pic
picGetPix(PIC *pic, int x, int y, PIX *pix);
picSetPix(PIC *pic, int x, int y, const PIX *pix);

// izmijeni boju piksela prema zadanoj funkciji;
// filter_id = 0 (ignore), 1 (grayscale) ili 2 (sepia)
pixFilter(PIX *pix, int filter_id);

#endif      /* _PICTURE_H_ */

```

Rješenje:

```

#include <picture.h>
int main(int argn, char *argv[]) {
    int x, y, w, h, tip = 1;
    PIC *p = picLoad(argv[1]);
    PIX d;

    w = picGetW(p);
    h = picGetH(p);

    if (!strcmp(argv[2], "sepia")) tip = 2;

    for (y = 0; y < h; ++y) {
        for (x = 0; x < w; ++x) {
            picGetPix(p, x, y, &d);
            pixFilter(&d, tip);
            picSetPix(p, x, y, &d);
        }
    }

    picSave(p, argv[3]);
    picFree(p);

    return 0;
}

```

10. [3] Napišite funkciju koja u predanom polju traži element s najvećim prosjekom varijabli a, b i c. Ukoliko takav element postoji, potrebno je vratiti pokazivač na njega (ako nije jedinstven, onda na bilo koji od takvih), a ukoliko takav element ne postoji, potrebno je vratiti NULL pokazivač. Struktura i prototip funkcije prikazani su u nastavku:

```
typedef struct {  
    const char *naziv; /* pokazuje na niz znakova s imenom grupe;  
                        NULL ako je kraj polja */  
    unsigned int a, b, c; // elementi  
} DAT;  
DAT* maxProsjek(DAT podaci[]);
```

Rješenje:

```
DAT* maxProsjek(DAT polje[]) {  
    int i, i_max;  
    float avg, max = -1.;  
  
    if (!polje[0].naziv) return NULL;  
  
    for (i = 0; polje[i].naziv; ++i) {  
        avg = (polje[i].a + polje[i].b + polje[i].c) / 3.  
        if (avg > max) {  
            i_max = i;  
            max = avg;  
        }  
    }  
  
    return &polje[i_max];  
}
```