

# Praktikum iz programiranja 2

## Materijal potreban za peti domaći zadatak

Detaljnija objašnjenja o svim postojećim formatima titlova se mogu pronaći uz pomoć Internet pretraživača.

U ovom dokumentu je priložen kratak i nepotpun opis tri formata za predstavljanje titlova na računaru, kao i primer titla sa istim tekstovima u tri različita formata.

Priložena su i uputstva za korišćenje funkcija standardne biblioteke programskog jezika C za očitavanje i ispisivanje proteklog i trenutnog vremena.

### Sadržaj

|                                                  |   |
|--------------------------------------------------|---|
| Ukratko o titlovima (subtitles)                  | 2 |
| Formati titlova                                  | 2 |
| SubRip (.SRT)                                    | 2 |
| MicroDVD (.SUB, .TXT)                            | 2 |
| MPlayer (.SUB)                                   | 3 |
| Funkcije programskog jezika C za rad sa vremenom | 3 |
| Često postavljana pitanja                        | 4 |

## Ukratko o titlovima (subtitles)

Sa povećanjem procesorske snage sa jedne, i multimedijalnih mogućnosti računara sa druge strane, pojava zapisa sa visokim kvalitetom kako pokretnih slika, tako i zvuka, bila je više nego očekivana. Tržište medijuma je ispratilo nove potrebe tržišta računara novim formatima koji pružaju veliki kapacitet za smeštanje podataka (razne varijante CD, DVD; MOD, ...). Bez obzira na sve ovo, za stvarno visok kvalitet multimedijalnog zapisa potrebno je i dalje relativno mnogo prostora, imajući u vidu trenutne kapacitete medijuma. Razvoj raznih formata kompresije sa minimalnim gubicima je još više olakšao prikaz multimedijalnih zapisa na kućnim računarima. Kako je neisplativo u zvaničnoj produkciji zapisa raditi prevođenje na sve moguće jezike, često se dešava da korisnici sami moraju praviti titlove na željenom jeziku i sa željenim vremenima pojavljivanja. Ovaj proces je često izvor velikog broja grešaka, kako sa aspekta tačnosti prevoda, tako i sa aspekta vremena pojavljivanja i uklanjanja titlova. Stvari se dodatno komplikuju ako se uzme u obzir i različit broj slika u sekundi od zapisa do zapisa (fps = frames per second), koji je najčešće (ali ne i uvek) 25 za PAL i 23.976 za NTSC zapise.

## Formati titlova

Postoji više formata titlova, koji imaju ili nemaju određene mogućnosti (višejezičnost, stil slova u tekstu titla, i slično). Sa aspekta ovog dokumenta, od pomenutih mogućnosti su mnogo bitniji načini zapisa vremena pojavljivanja i uklanjanja titlova. Tri prikazana formata koriste kao parametre apsolutno vreme od početka zapisa, ili relativno vreme od prethodnog titla, ili broj slika od početka zapisa. Osim navedenog, bitna razlika između navedenih formata je i način predstavljanja prelaska teksta u narednu liniju – SubRip i MPlayer koriste standardni '\n' (prelazak u naredni red), dok MicroDVD koristi '|' (engl. *pipe*). Prilikom konverzije između dva formata, treba voditi računa o navedenim razlikama koje mogu postojati u načinu predstavljanja vremena i tekst.

## SubRip (.SRT)

Ovaj format karakteriše apsolutno vreme pojavljivanja i uklanjanja titlova, sa preciznošću od 1 milisekunde, uz razdvajanje dva titla jednim praznim redom teksta. Primer za ovaj format sledi:

```
1
00:00:15,000 --> 00:00:18,000
A long, long time ago...

2
00:00:18,000 --> 00:00:21,000
in a galaxy far away...

3
00:00:21,000 --> 00:00:24,000
Naboo was under an attack.

4
00:00:25,000 --> 00:00:27,500
And I thought me and
Qui-Gon Jinn could

5
00:00:27,500 --> 00:00:30,000
talk the Federation into

6
00:00:30,000 --> 00:00:34,000
...maybe cutting them a
little slack.
```

## MicroDVD (.SUB, .TXT)

Ovaj format karakteriše redni broj slike (engl. *frame*) u kome titl treba da se pojavi, odnosno ukloni. Zavisno od broja slika u sekundi za dati multimedijalni zapis (engl. *frame per second*, odnosno *fps*),

ovaj titl se može pojaviti ranije (fps=25), odnosno kasnije (fps=23.976). Svaki titl je u jednom redu teksta. Primer za ovaj format, koji sadrži isti titl, kao i prethodni, je prikazan ovde, uz pretpostavku da je fps=25:

```
{375}{450}A long, long time ago...
{450}{525}in a galaxy far away...
{525}{600}Naboo was under an attack.
{625}{688}And I thought me and|Qui-Gon Jinn could
{688}{750}talk the Federation into
{750}{850}...maybe cutting them a|little slack.
```

## MPlayer (.SUB)

Za razliku od prethodna dva formata, koji vreme uglavnom računaju od početka zapisa, ovaj format ima i mogućnost da računa vreme relativno – svaki titl ima informaciju o vremenu proteklom u odnosu na prethodni titl, i o sopstvenom trajanju. Prvi broj označava koliko treba čekati u odnosu na uklanjanje prethodnog titla, a drugi broj označava koliko dugo dati titl treba da bude prikazan. Nakon reda sa brojevima, sledi jedan ili dva reda sa tekstom titla. Kao i kod prvog opisanog formata, i ovde je razdvajanje dva titla ostvareno jednim praznim redom teksta. Primer sledi:

```
15 3
A long, long time ago...

0 3
in a galaxy far away...

0 3
Naboo was under an attack.

1 2.5
And I thought me and
Qui-Gon Jinn could

0 2.5
talk the Federation into

0 4
...maybe cutting them a
little slack.
```

## Funkcije programskog jezika C za rad sa vremenom

Operativni sistem vodi evidenciju o utrošenom procesorskom vremenu za svaki od trenutno aktivnih procesa. U programskom jeziku C, ovo vreme se može dobiti pozivom funkcije `clock()` i izraženo je u „otkucajima“ sistemskog časovnika (engl. *clock ticks*). Svaka implementacija jezika C podrazumeva sopstvenu vrednost simboličke konstante `CLOCKS_PER_SEC` (u ranijim verzijama standarda `CLK_TCK`), koja predstavlja broj „otkucaja“ sistemskog časovnika u toku jedne sekunde.

Sa druge strane, u svakom trenutku je moguće od operativnog sistema dobiti informaciju o vremenu u sistemskom časovniku, kako i o datumu i vremenskoj zoni gde se dati računar nalazi. Jednom kada tačno vreme bude očitano, ne postoji nikakva prepreka da se ono pretvori u odgovarajuću strukturu, odnosno prikaže na željeni način preko proizvoljnog znakovnog niza. Priloženi programski segment ilustruje prikaz tačnog vremena bez prikaza datuma:

```
#include <time.h>
...
time_t pIntTime = time(NULL);
struct tm* currentLocalTime = localtime(&pIntTime);
char* dateTimeString = calloc(100+1, sizeof(char));
/* if time setting and memory allocation was successful, create complete message */
if (currentLocalTime && dateTimeString)
    /* format the time as needed */
    strftime(dateTimeString, 100, "%H:%M:%S", currentLocalTime);
```

Dodatne informacije o ovde navedenim funkcijama (i svim ostalim koje se nalaze u zaglavlju

`time.h`) se mogu naći u zbirci prof. Krausa i na raznim dokumentacionim stranicama (kako onima koje dolaze uz razvojno okruženje, tako onima dostupnim na Internetu).

## Često postavljana pitanja

**Q:** Šta su ulazni parametri programa?

**A:** Program dobija imena datoteka i pomeraj u vremenu preko parametara komandne linije. Ove parametre korisnik zadaje razvojnom okruženju **Visual C++ 6.0** na sledeći način: stavka **Project** iz glavnog menija, zatim stavka **Settings** (ili Alt+F7). Dijalog koji će biti prikazan ima, između ostalih, i karticu **Debug**. Kategorija **General**, koja će standardno biti prva otvorena, ima mesto predviđeno za argumente komandne linije - **Program Arguments**. Tu treba uneti sve ono sto bi se u komandnoj liniji otkucalo posle imena programa. Na primer, ako imamo "HH-eng.srt HH-eng2.srt 32000", to znači da je ulazna datoteka HH-eng.srt, izlazna datoteka HH-eng2.srt, te da svi titlovi iz ulazne datoteke, pre snimanja u izlaznu datoteku, treba da budu pomereni 32000 milisekunde kasnije u odnosu na original. Slično je i u okruženju **Visual C++ 2005**: stavka **Project** iz glavnog menija, zatim stavka **Properties** (ili Alt+F7). Dijalog koji će biti prikazan ima sa leve strane kontrolu sa raznim stavkama. Treba izabrati stavku **Configuration Properties** i unutar nje **Debugging**. Sa desne strane će biti prikazano nekoliko opcija, od kojih je druga **Command Arguments** i ima istu namenu kao **Program Arguments** u **Visual C++ 6.0**.

**Q:** Kakvo ažuriranje podataka u vremenu program treba da radi?

**A:** Treba da pomeri sve titlove napred ili nazad u vremenu, zavisno od vrednosti odgovarajućeg parametra komandne linije.

**Q:** Sta znači pomeraj titla u vremenu? Da li prilikom prebacivanja titlova iz jedne u drugu datoteku svaki titl treba da bude pomeren (npr: 56.78 sekundi unapred ili unazad), pa će shodno tome celokupni sadržaj izlazne datoteke kasniti ili žuriti u odnosu na original?

**A:** Upravo tako.

**Q:** Kako da obezbedim uslovno prevođenje? Koristim `#ifdef`, `#elif` i sve ostale potrebne direktive, ali kako da definišem željeni simbol, a da ne koristim `#define PERA` koje moram da brišem kada ne želim da `PERA` bude definisano?

**A:**

**Visual C++ 6.0:** Prvo iz menija **Build** odaberite stavku **Configurations**, te u datom dijalogu dodajte (**Add...**) novu konfiguraciju baziranu na postojećoj `Win 32 Debug` konfiguraciji, i dajte joj ime po želji. Zatvorite dijalog i idite na **Project->Settings** (Alt+F7). Tamo odaberite konfiguraciju koju želite da menjate, a zatim idite na karticu **C/C++**. Kategorija **General** sadrži stavku **Preprocessor Definitions**, gde treba dodati željeni simbol koji ste u tekstu programa koristili u `#ifdef` ili `#ifndef` direktivi preprocesoru. Kada želite da prevedete program korišćenjem date konfiguracije, koja sadrži ili ne sadrži odgovarajuće simbole, iz menija **Build** odaberite stavku **Set Active Configuration** i tu podesite željenu konfiguraciju.

**Visual C++ 2005:** Prvo iz menija **Build** odaberite stavku **Configuration Manager**, te u datom dijalogu izaberite listbox kontrolu koja pokazuje postojeću `Debug` konfiguraciju i ispod postojećih konfiguracija pronađite opciju `<New...>` i izaberite je. U sledećem dijalogu novoj konfiguraciji dajte ime po želji i obavezno u listbox kontroli `Copy settings from` izaberite postojeću `Debug` konfiguraciju. Zatvorite dijalog i idite na **Project->Properties** (Alt+F7). Tamo odaberite konfiguraciju koju želite da menjate. Dijalog koji će biti prikazan ima sa leve strane kontrolu sa raznim stavkama. Treba izabrati stavku **Configuration Properties** i unutar nje **C/C++**, unutar koje postoji **Preprocessor**. a desne strane će biti prikazano nekoliko opcija, od kojih je prva **Preprocessor Definitions** i ima istu namenu kao u **Visual C++ 6.0**. Tu treba dodati željeni simbol koji ste u tekstu programa koristili u `#ifdef` ili `#ifndef` direktivi preprocesoru. Kada želite da prevedete program korišćenjem date konfiguracije, koja sadrži ili ne sadrži odgovarajuće simbole, u **Build-> Configuration**

**Manager** odaberite željenu konfiguraciju.