**UNIVERZITET U BEOGRADU**

**ELEKTROTEHNICKI FAKULTET**

***Katedra za elektroniku***

***Predmet: Racunarska elektronika***

****

**PROJEKAT 2015/2016**

Predmetni profesor: **Dr Milan Prokin**

Predmetni asistent: **Aleksandra Lekic**

Projekat radili:

|  |  |  |
| --- | --- | --- |
| **Ime** | **Prezime** | **broj indeksa** |
| **Aleksandra** | **Skrbic** | **0163/2012** |
| **Selena** | **Colovic** | **0084/2012** |

Datum: 15.06.2016. (last revision)

**SADRZAJ**

1. **Postavka zadatka ............................................................................................................. 3**
2. **Koncept realizacije programa ....................................................................................... 4**
3. **Realizacija (izvorni kod) ................................................................................................ 6**

* **AddTwo.asm (ostavljen je ovaj naziv)............................................................... 6**

1. **Zakljucak ....................................................................................................................... 12**
2. **Literatura ...................................................................................................................... 13**

**1. POSTAVKA ZADATKA**

Projektni zadatak koji nam je bio dodeljen, je pod rednim brojem 15 na listi projekata. Trebalo je uraditi sledece: dozvoliti korisniku da unese ime fajla sa ekstenzijom **.pls** u prozoru konzole, a potom je bilo neophodno izvrsiti izvesnu obradu tj. promenu formata fajla i cuvanje rezultata obrade u izlazni fajla sa extenzijom **.m3u**. Na sajtu je dat primer izgleda obe liste. I konkretno u projektu koristila sam bas te liste za validaciju funkcionalnosti koda. U nastavku ce biti dati primer prikaza **.pls** ulazne liste i **.m3u** izlazne liste u **Notepad++**.

**Primer intput fajla:**

**[playlist]**

**NumberOfEntries=50**

**File1=**nesto-nesto.mp3

**Title1=**nesto- nesto.mp3

**Length1=**1

**File2=**E:\Muzika\Dream Theater\Falling Into Infinity\Dream Theater\_Falling Into Infinity\_01\_New Millennium.mp3

**Title2=**Dream Theater - New Millennium

**Length2=**500

**File3=**E:\Muzika\Dream Theater\Falling Into Infinity\Dream Theater\_Falling Into Infinity\_02\_You Not Me.mp3

**Title3=**Dream Theater - You Not Me

**Length3=**298

**Primer output fajla:**

**#EXT3M3U**

**#EXTINF:**1**,**nesto- nesto.mp3nesto-nesto.mp3**#EXTINF:**500**,**Dream Theater - New MillenniumE:\Muzika\Dream Theater\Falling Into Infinity\Dream Theater\_Falling Into Infinity\_01\_New Millennium.mp3**#EXTINF:**298**,**Dream Theater - You Not MeE:\Muzika\Dream Theater\Falling Into Infinity\Dream Theater\_Falling Into Infinity\_02\_You Not Me.mp3

**2. KONCEPT REALIZACIJE PROGRAMA**

Dakle, u okviru ovog poglavlja objasnicu ideju za realizaciju samog programa za zahtevanu funkcionalnost.

Najpre je bilo potrebno dozvoliti korisniku da unese ime ulaznog fajla u prozoru konzole (**.pls** fajl). To omogucavamo pozivom fje ***ReadString*** za niz unet od strane korisnika. Ime koje korisnik unosi cuvacemo u ***srcFilename*** stringu i to cemo kasnije obraditi, da bismo dobili isto ime za izlazni fajl, samo sa drugom extenzijom (**.m3u**). Zatim, pozivom fje ***OpenInputFile***,otvaramo fajl ukoliko on postoji. Sada, sledi provera validnosti unetog imena tj. postojanja ulaznog fajla: to se odnosi na **fileHandle**, odnosno mogucnost ispravnog otvaranja fajla (fajl postoji i nije prazan). Kada je ta provera prosla, ucitavamo ulazni fajl u bafer koji smo inicijalizovali u **.data** segmentu koda. Pozivamo fju ***ReadFromFile***. Odrediste ce biti pomenuti bafer, a sa fajlom komuniciramo preko **fileHandle**-a. Takodje, vrsimo proveru velicine bafera za ulazni fajl i ispis odgovarajuce poruke ukoliko greska prilikom izbora velicine postoji. Ukoliko je velicina bafera odgovarajuca, prikazujemo je u **GUI** prozoru. Potom sam izvrsila ispis vrednosti smestenih u baferu takodje, radi provere validnog citanja svih podataka iz ulaznog fajla. Za kraj zatvaramo ulazni fajl, jer nam vise ne treba; ucitan je u bafer.

Sada, sledi kompleksniji deo projekta, a to je obrada sadrzaja pomenutog bafera i cuvanje rezultata u izlazni bafer. Najpre je uz pomoc instrukcije bez operanada ***cld***definisano da ce se stringu pristupati u inkrementirajucem poretku.

Ideja je bila sledeca: citamo u baferu podatke sve dok ne dodjemo do "=". Potom preskacemo taj red (taj red nam za obradu ne treba), i dolazimo do reda koji sadrzi, za nasu obradu, bitne podatke. Taj red citamo do kraja. Ovo radimo koristeci registre za *Source Index* (**ESI**) on je pointer na elemente bafera sa ulaznim fajlom. U **ACC** (low part, **AL**) stavljamo stringove koji su nam od znacaja: "=" i 0dh sto je simbol za prelazak u novi red. Kada ocitamo red koji nam je neophodan za dalju obradu, **ESI** vrednost cuvamo da znamo gde smo stali u obradi ulaznog fajla, a **ACC** koristimo kao registar koji nam moze pomoci da se obracamo podacima koji su nam u ovom redu znacajni, i potom ih stavljamo na stek (**ESP** - stek pointer). Kada se pogleda trazeni format liste na izlazu, kao i same kke steka, tj. kako raste i na sta ukazuje, citacemo karaktere unazad sve do "=". Sada smo na stek stavili string koji predstavlja putanju do date numere playlist-e. Potom nastavljamo da citamo tamo gde smo stali (**ESI**). Citamo sve do kraja sledeceg reda. Ponovo idemo sa **ACC** registrom kao pointerom unazad, sve dok ne dodjemo do "=". Do tog trenutka smestamo karaktere na stek, tako da sada kada zavrsi ovu petlju, program je na stek smestio putanju, ali i naziv numere u listi. Potom nastavljamo sa daljom obradom i citamo ceo sledeci red (tj. do kraja sledeceg stringa) u okviru koga je sakriven podatak o trajanju numere. Ovde ce postojati i specijalan slucaj kada necemo ocitavati znak za novi red (0dh) nego 0h (**EOF**). U tom slucaju, nakon upisa podatka na stek, samo je potrebno dekrementirati **ESI** i to ne zapisujemo na stek vec samo prelazimo na ispis u izlazni bafer. U protivnom, ukoliko je klasicno procitan novi red, takodje smestimo podatak o vremenu na stek citajuci unazad do "=" i nakon toga prelazimo na ispis u izlazni bafer.

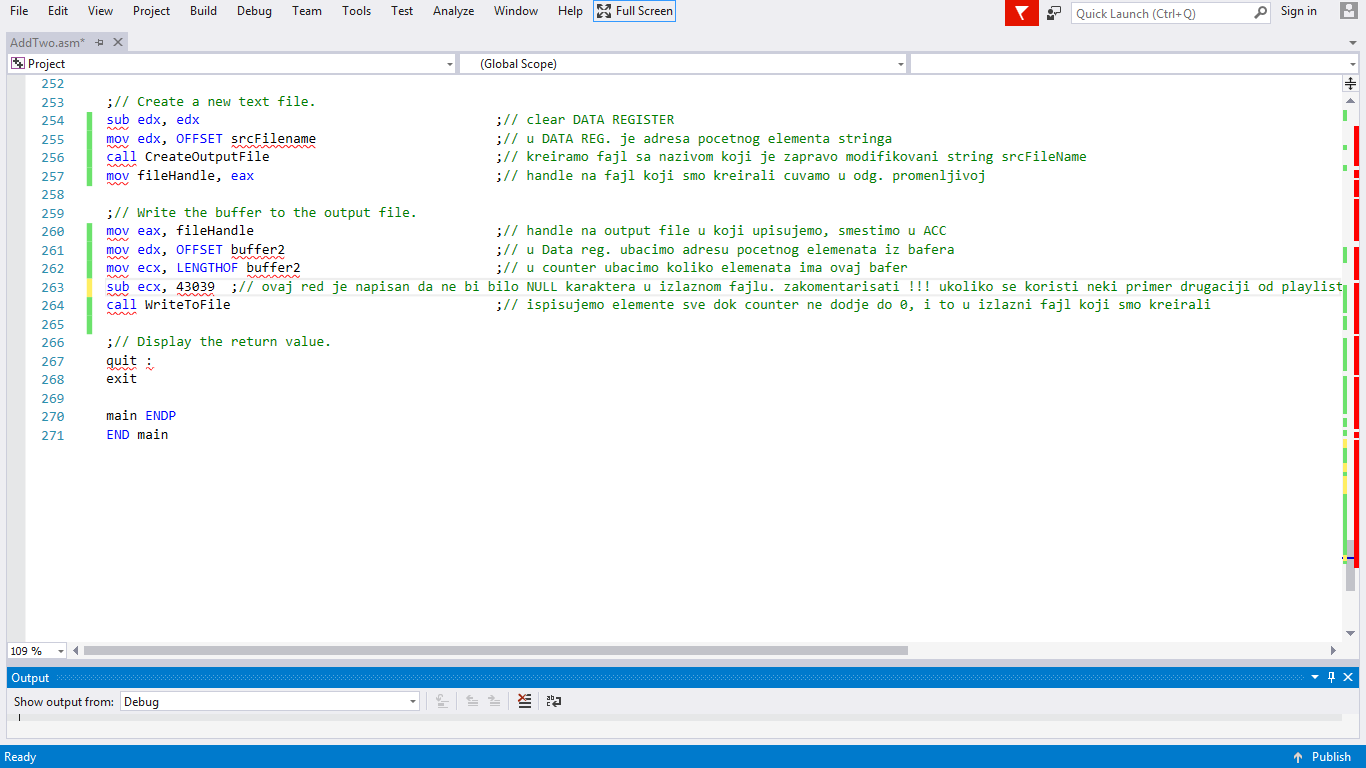
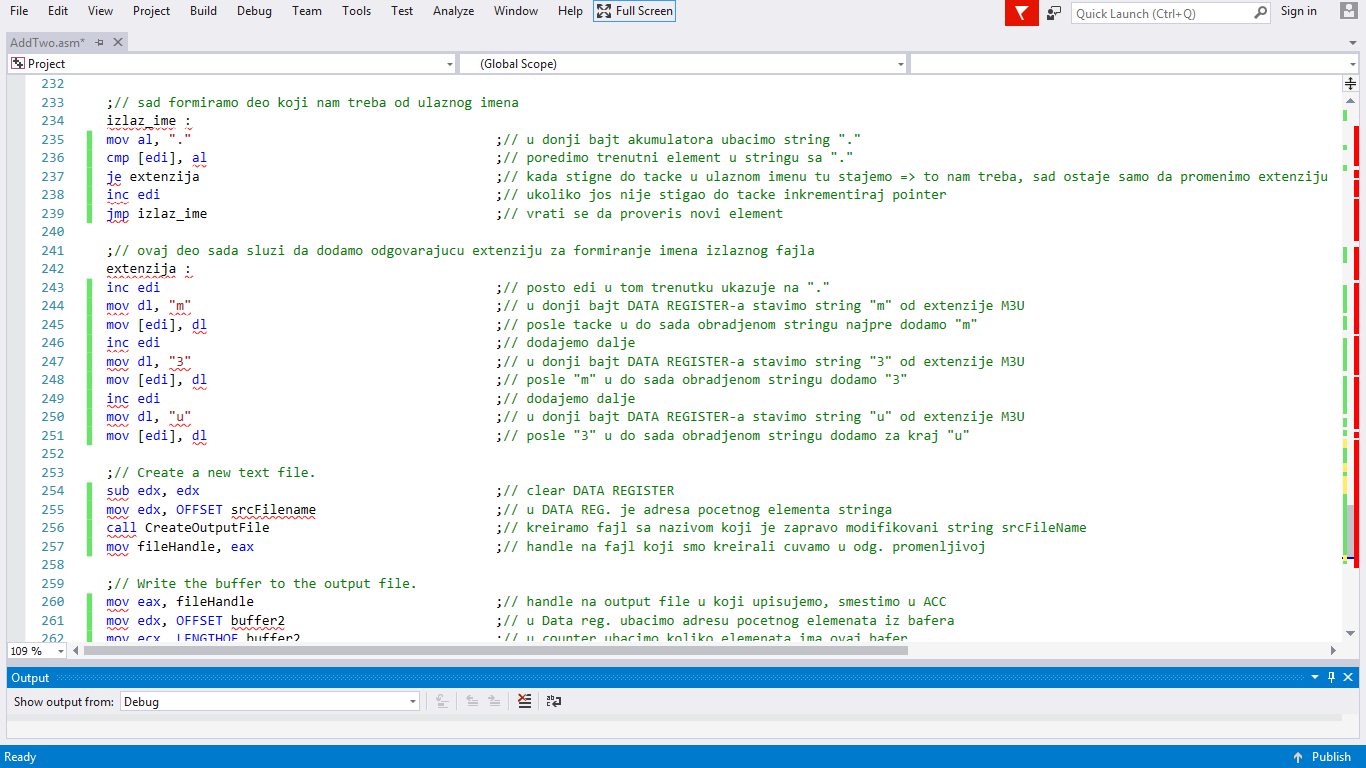
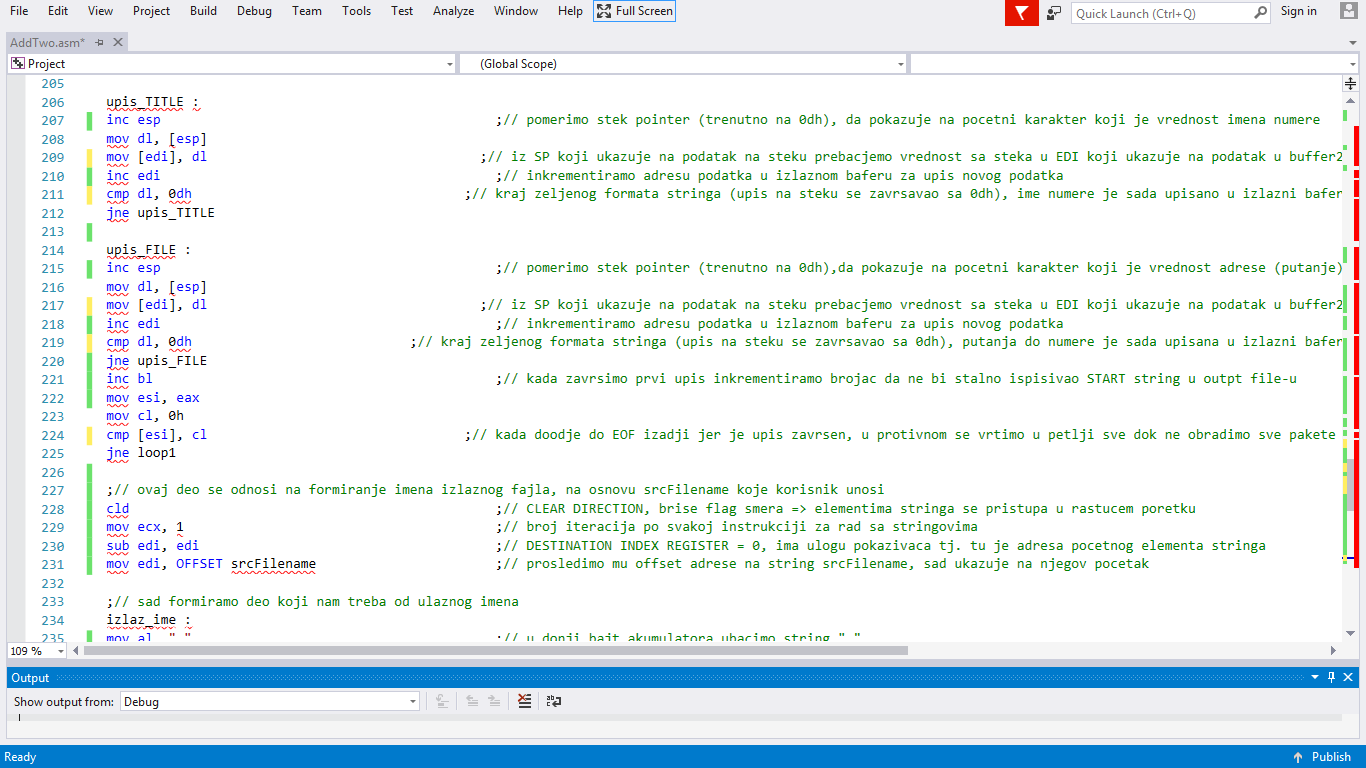
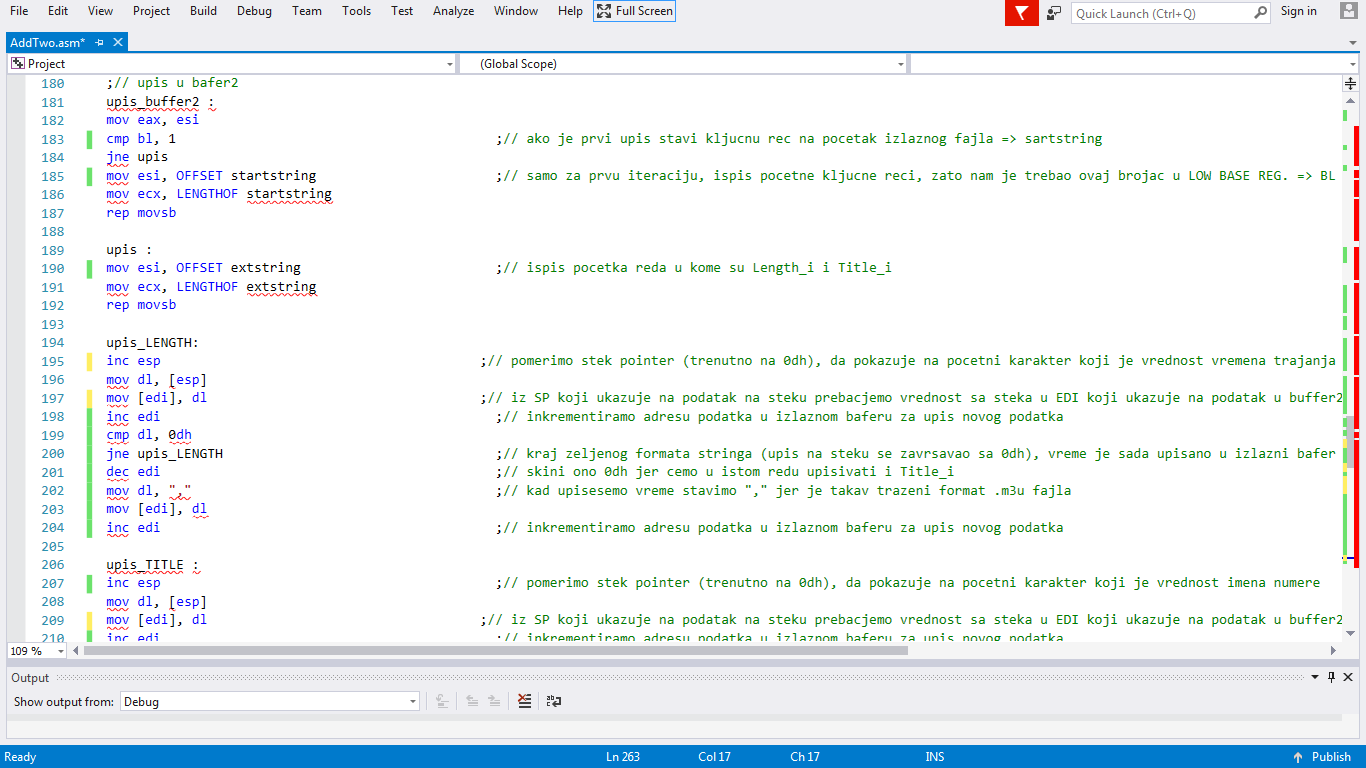
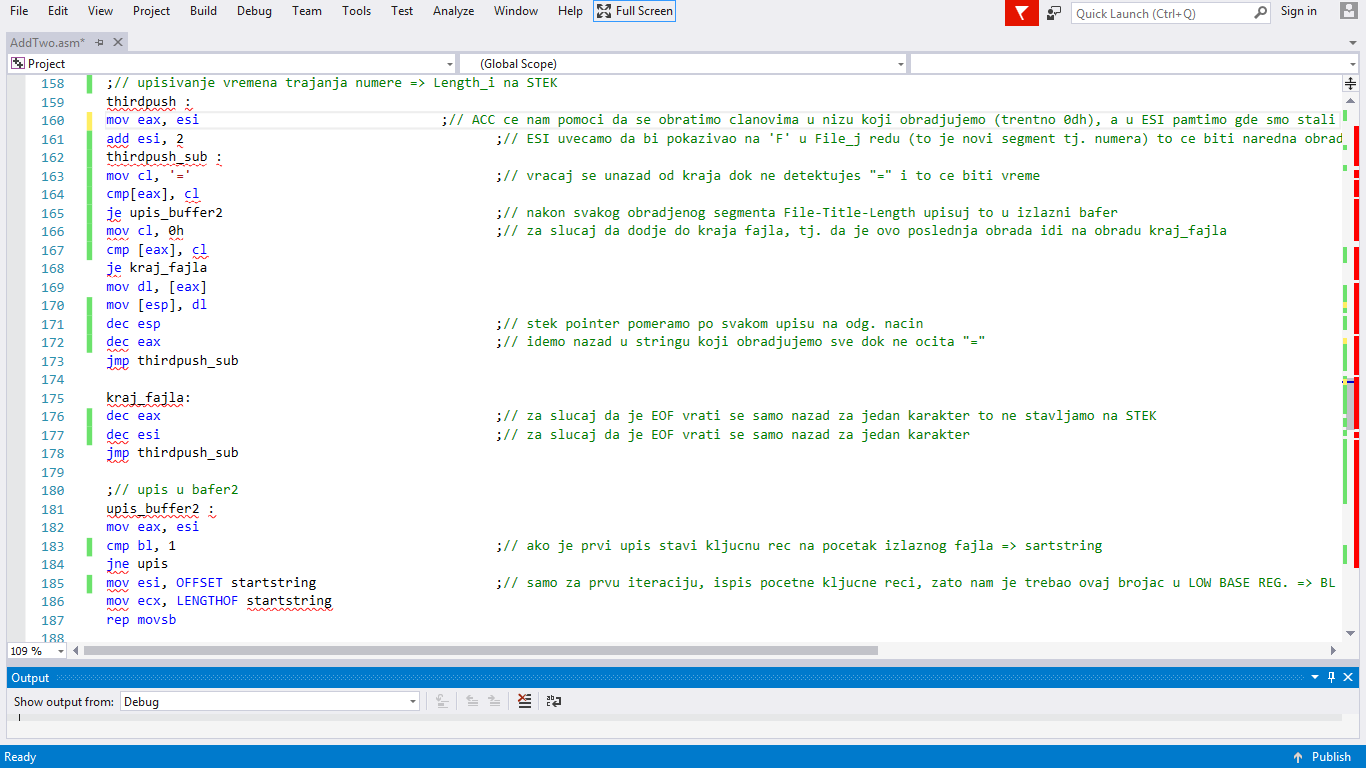
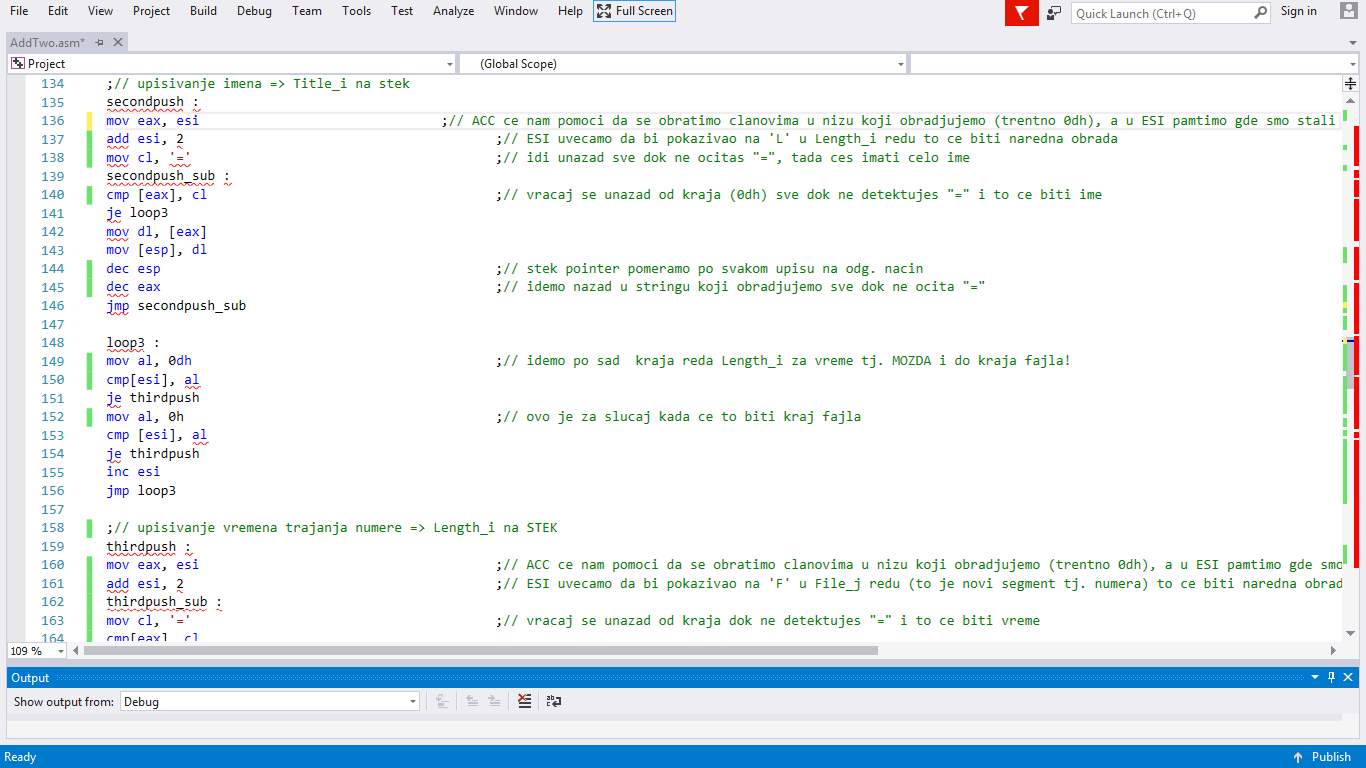
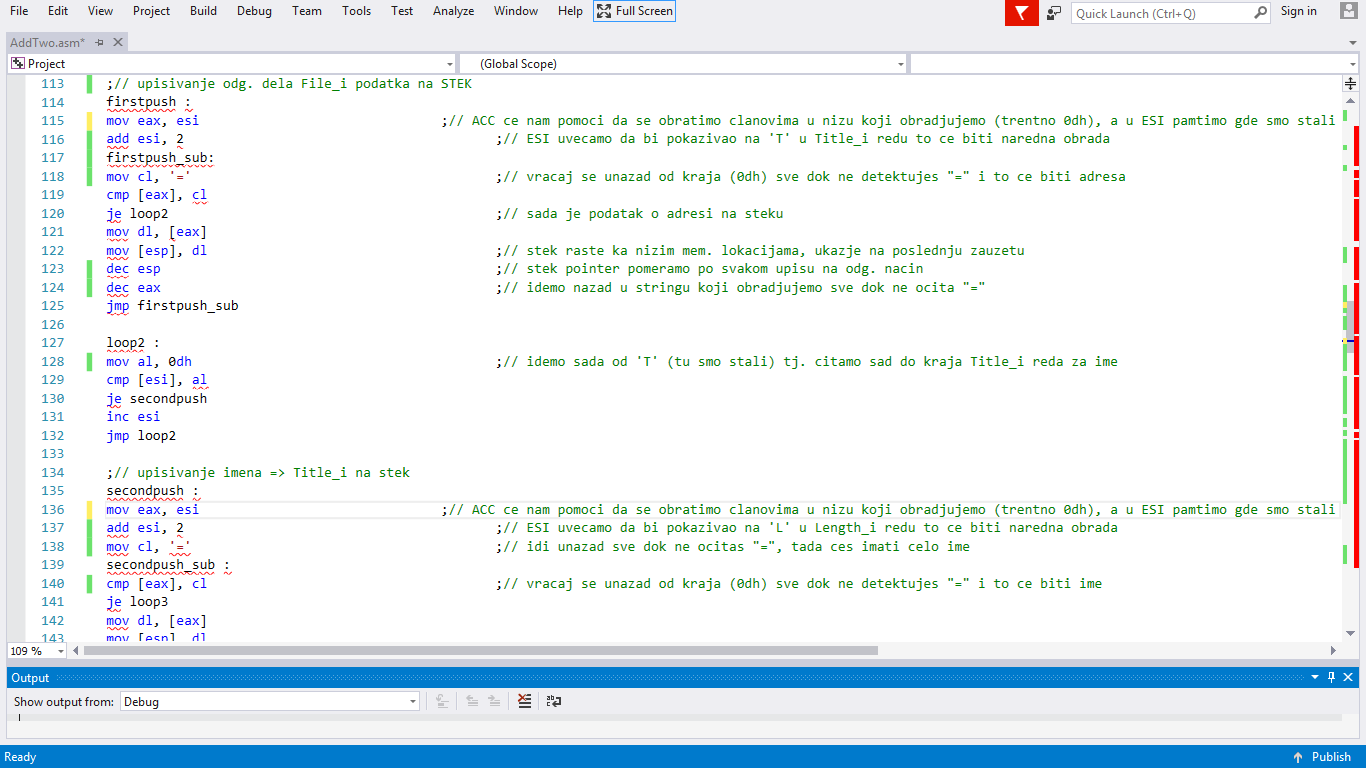
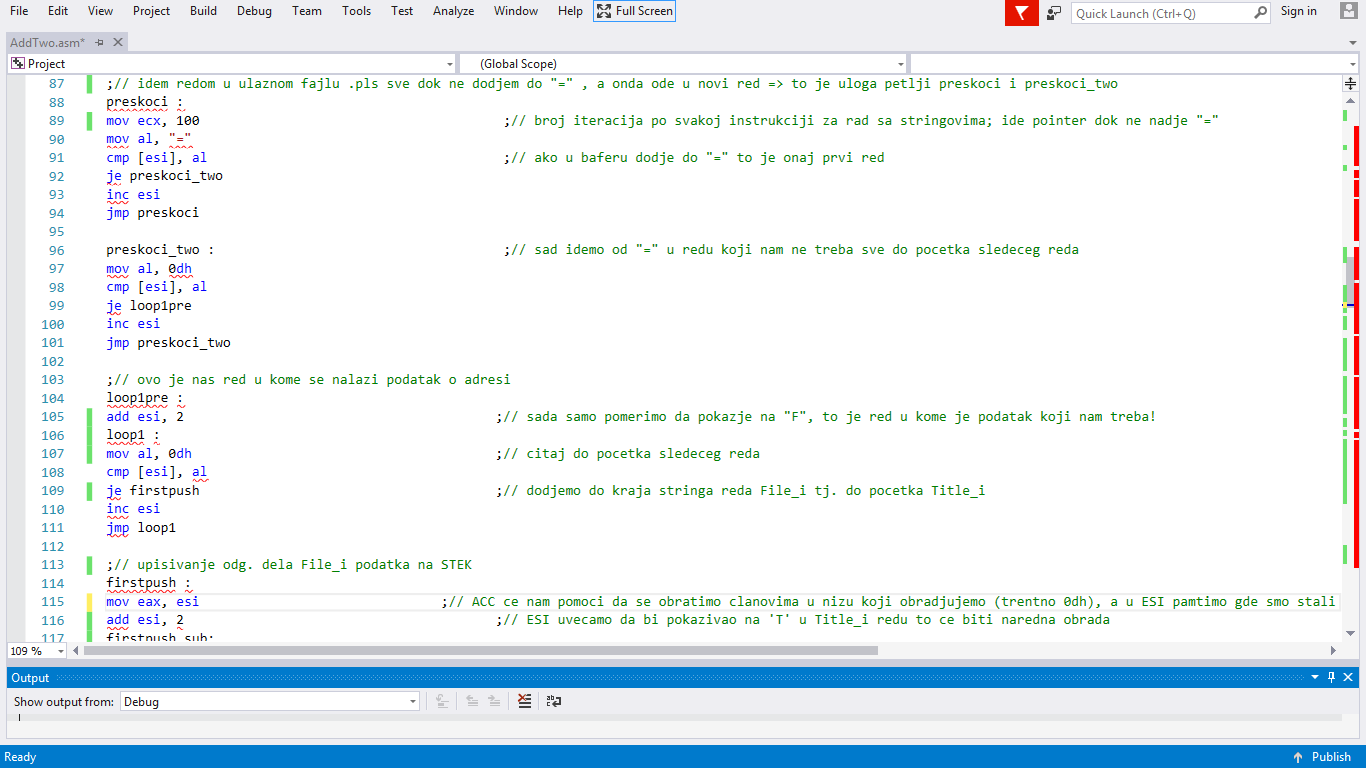
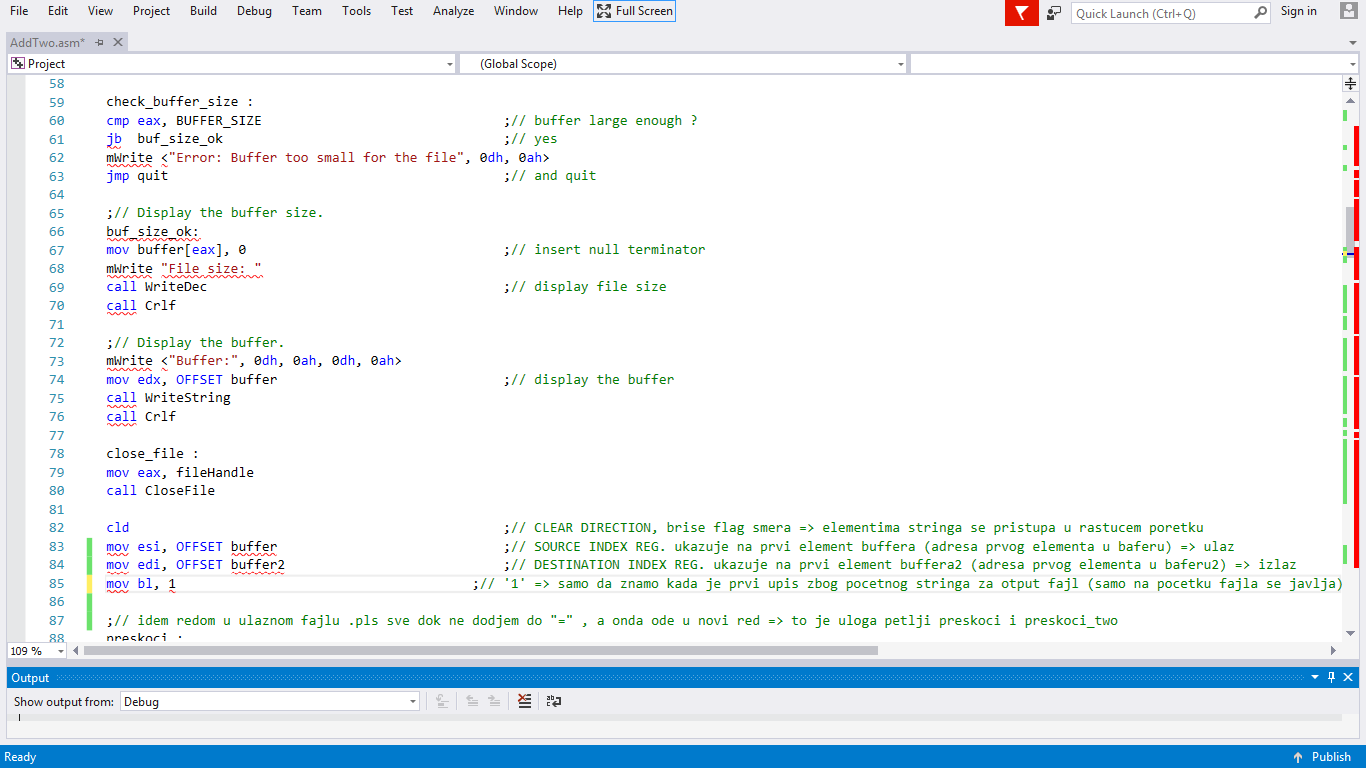
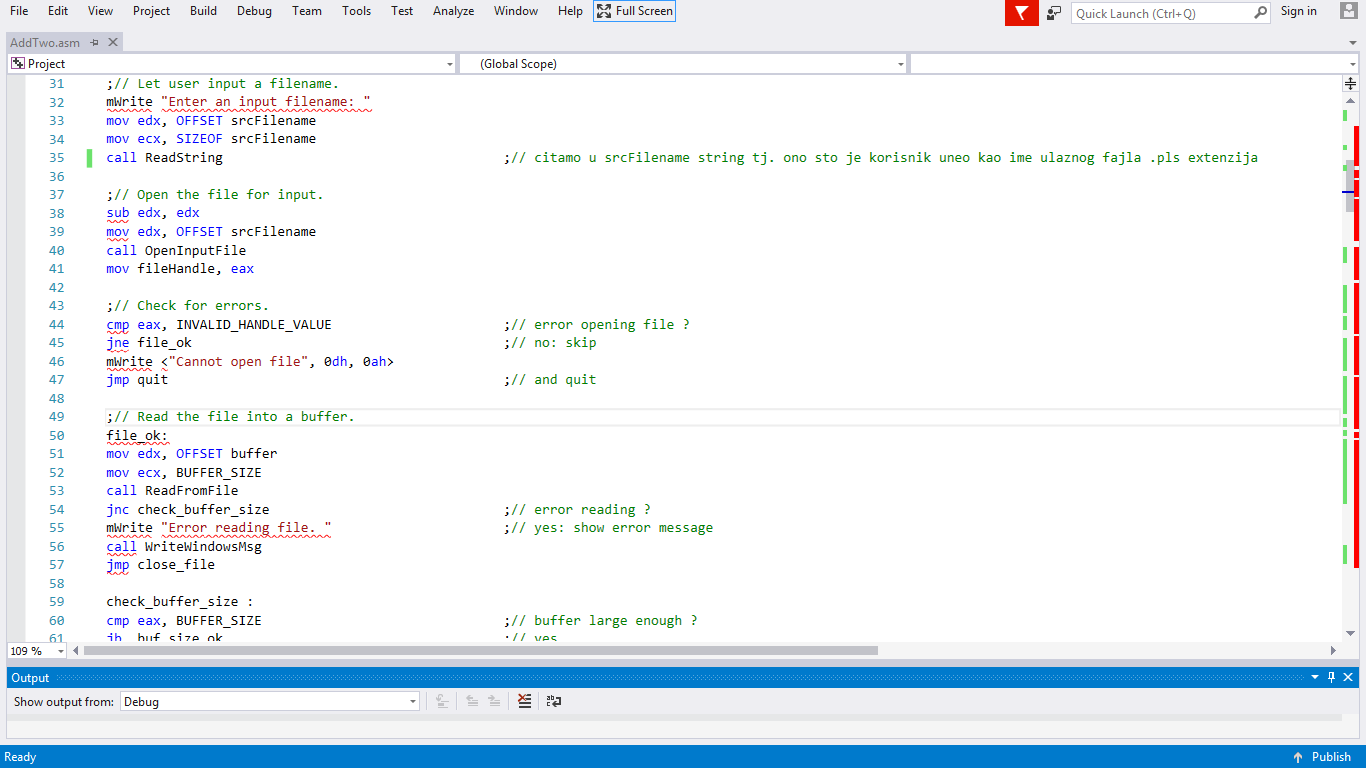
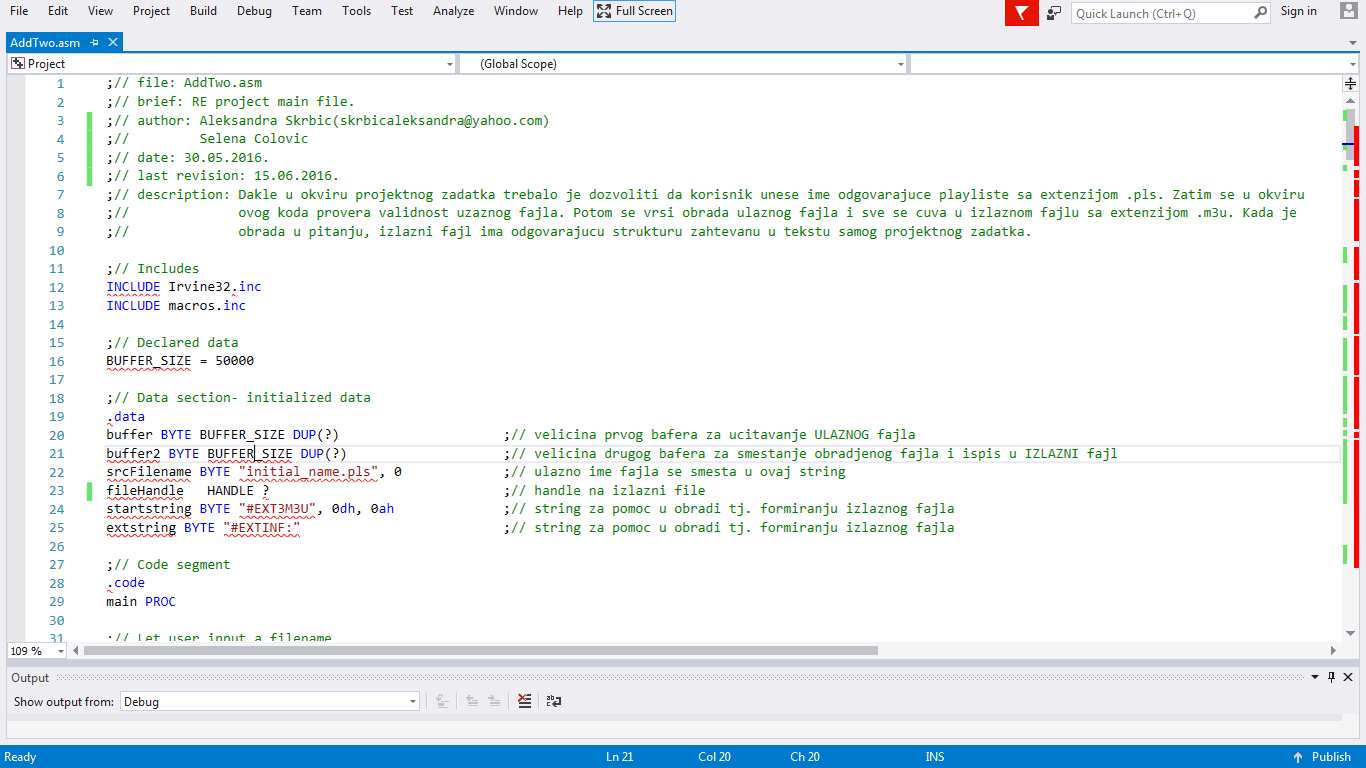
Dakle svaki put nakon smestanja vrednosti u odgovarajucoj formi za svaku numeru liste: **File**, **Title** i **Length**, mi cemo odmah sa steka sve upisivati u izlazni bafer. Ukoliko je prva iteracija (pocetak izlaznog fajla), a za to cemo imati pomocnu promenljivu tj. brojac (**BL**) da nam javi da je pitanju prvi upis, imacemo upis kljucne reci, smestene ***startstring***,koju cemo ispisivati u izlaznom baferu. Potom ce slediti standardni upis za svaku numeru liste. Najpre ispisjemo string ***extstring*** na pocetk reda, zatim sa steka skidamo podatak o poslednjoj upisanoj vrednosti za **Length** (preskacemo ono 0dh sa kojim se string zavrsavao), i smestamo ga u izlazni bafer. Potom upisujemo ",". Nakon toga na slican nacin skidamo sa steka vrednost za **Title** numere. Sada i nju smestamo u izlazni bafer. Potom prelazimo na novi red i tu upisujemo novu vrednost koju skidamo sa steka, a to je preostala **File** vrednost za numeru. Sada smo kompletno izvrsili obradu ulaznih vrednosti na odgovarajuci nacin za jednu numeru u listi. Sada samo inkrementiramo brojac **BL**, da oznacimo da nema vise potrebe za ***startstring***-om. Potom proveravamo da li je **EOF**.Ukoliko nije ovu obradu vrsimo za svaku narednu numeru u playlist-i. Ukoliko smo dosli do kraja fajla, potrebno je da kreiramo izlazni fajl u koji cemo smestiti sve rezultate obrade iz bafera2.

Ponovo sam izvrsila pozive odgovarajucih instrukcija za settovanje rada sa stringom. Zatim je usledilo formiranje imena za izlazni fajl, tako sto sam citala string formiran na osnovu unosa korisnika u prozoru, ***srcFilename*** i to sve dok nisam dosla do ".". Zatim je samo promenjena extenzija ulaznog fajla prepisivanjem na mesto "**pls**", "**m3u**". Na ovaj nacin imali smo validno ime fajla koje je zahtevano u textu projekta.

Za sam kraj, prosledjivanjem odgovarajcih parametara odnosno smestanjem odg. vrednosti u registre i pozivom fja ***CreateOutputFile***, kao i ***WriteToFile***krairan je output fajl odgovarajuceg imena i extenzije, i u njega je upisan sadrzaj izlaznog bafera (buffer2).

Time je trazena funkcionalnost u potpnosti zadovoljena. Kod je testiran na ulaznoj ***playlist.pls***, a rezultat je bila izlazna lista ***playlist.m3u***, koja je u potpunosti identicna zahtevanoj formi u textu projektnog zadatka. Obe su prilozene u **.zip** datoteci.

**3. REALIZACIJA (IZVORNI KOD)**

****

\* Dodala sam slike za izvorni kod, iz razloga sto se po mom misljenju to cini preglednijim od jednostavnog kopiranja celog koda. Ovde se na slikama moze videti i kako je zakomentarisana svaka linija koda, tj. tu se veoma detaljno objasnjava sta svaki red koda predstavlja. Ukratko, sve je objasnjeno u prethodnom poglavlju izvestaja.

**4. ZAKLJUCAK**

Ono sto sam samo zelela da napomenem je da je celokupni dizajn koda u skladu sa zahtevima u projektom zadatku, i da je funkcionalnost u potpunosti ispunjena. Medjutim, jedini problem koji eventualno moze da se uoci je sledeci: NIJE MOGUCE TACNO PODESITI VELICINU BAFERA (izlaznog). Ulazni i izlazni bafer ne mogu se staviti na istu velicinu, kada je smestanje podataka u pitanju, iz prostog razloga sto su formati ulaznog i izlaznog fajla drugaciji. Zato ce se javiti jedan problem koji sam primetila na samom ispisu u izlazni fajl. Za sva neiskoriscena mesta prilikom obrade i upisa podataka iz ulaznog fajla, pisace se u izlaznom fajlu simbol **NULL**! Zato u kodu postoji jedan red koji umanjuje velicinu bafera za odredjeni broj, da bi velicina bila potpunosti odgovarajuca za ulaznu playlist-u datu kao primer za ovaj projektni zadatak. Tako da posto je ovaj program na njoj testiran, on u potpunosti i podesava velicinu izlaznog bafera specijalno za tu listu. Ukoliko zelite da isprobate na nekoj drugoj listi tipa **.pls**, taj red, kao sto i pise u samom kodu **zakomentarisite**. To je jedini problem koji sam uocila u realizaciji resenja. Probala sam da ga resim upisom **EOF** simbola (**0h**) na kraju, posle upisa poslednjeg karaktera iz bafera, ali to nije uspelo. Pokusala sam i brojac da kreiram, pa da velicinu buffera2 cuvam u nekoj promeljivoj, pa da kad se ona "prevrti" prekinem upis, ali ni to nije uspelo. Tako da, to bi to bila eventualna mala mana, jer bi izlazni fajl imao te **NULL** karaktere (simbole) u polsednjem redu, ali sve u svemu **trazena funkcionalnost je u potpunosti zadovoljena**.

**5. LITERATURA**

[1] Materijal sa vezbi - Aleksandra Lekic

[2] Assembly Language for x86 Processors, 7th edition by Kip Irvine, Florida International University (chapters I found on the Internet)

KRAJ :)