

ARM

Napomena: poglavlja obojena crvenom bojom ne će
vam trebati

Sadržaj

1. Opis zipa	8
1.1. a.s	8
1.2. ARM.pdf.....	8
1.3. ARMPluginInterfaces.dll	8
1.4. ARMSim.exe	8
1.5. ARMSim.exe.config.....	8
1.6. ARMSim.Plugins.EmbestBoardPlugin.dll	8
1.7. ARMSim.Plugins.UIControls.dll	8
1.8. ARMSimSamplePlugins.vsi	8
1.9. ARMSimWindowManager.dll	8
1.10. ARMSim_UserGuide4Plus.pdf	8
1.11. DockingWindows.dll	8
1.12. DotNetMagic2005.dll	8
1.13. Primjer 1. Pregled nekih načina naslovljavanja.s	8
1.14. Primjer 2. Čitanje sa stdina i ispis na zaslon.s	8
1.15. Primjer 3. Otvaranje datoteke i čitanje iz nje.s	8
1.16. Primjer 4. Blok if-then-else.s	8
1.17. Primjer 5. Petlja for.s	8
1.18. Primjer 6. Petlja while.s	8
1.19. StaticWindows.dll	9
1.20. stdin.txt	9
2. Opis simulatora.....	9
2.1. Glavni izbornik	9
2.1.1. File [Datoteka]	9
2.1.1.1. Load [Učitaj]	9
2.1.1.1.1. Standardni okvir za otvaranje datoteka	9
2.1.1.2. Load Multiple [Učitaj više]	10
2.1.1.3. Reload [Ponovno učitaj]	10
2.1.1.4. Preferences [Preferencije]	10
2.1.1.5. (do 4 posljednje otvarane datoteke)	10
2.1.1.6. Exit [Izidi]	10
2.1.2. View [Prikaz]	11
2.1.2.1. Registers [Registri].....	11

2.1.2.2. Output [Izlaz]	11
2.1.2.3. Stack [Stog]	11
2.1.2.4 Watch [Motrenje].....	11
2.1.2.5. Data Cache [Podatkovna priručna memorija]	11
2.1.2.6. Instruction Cache [Naredbena priručna memorija]	11
2.1.2.7. Unified Cache [Ujedinjena priručna memorija]	11
2.1.2.8. PluginsUI [Korisničko sučelje priključaka]	11
2.1.2.9. Memory [Memorija]	11
2.1.3. Cache [Priručna memorija]	11
2.1.4. Debug [Uklanjanje pogrešaka]	12
2.1.4.1. Run [Izvršavaj]	12
2.1.4.2. Step Into [Korakni unutra]	12
2.1.4.3. Step Over [Korakni preko]	12
2.1.4.4. Restart [Ponovno pokreni]	12
2.1.4.5. Stop [Zaustavi]	12
2.1.4.6. Clear All Breakpoints [Očisti sve prekidne točke].....	12
2.1.5. Watch [Motrenje]	12
2.1.5.1. Add Watch [Dodaj motrenje]	13
2.1.5.1.1. Files [Datoteke]	13
2.1.5.1.2. Labels [Labele]	13
2.1.5.1.3. Display As [Prikaži kao]	13
2.1.5.1.4. Integer Format [Cjelobrojno obličje]	13
2.1.5.1.5. Base [Baza]	13
2.1.5.1.6. OK [U redu]	13
2.1.5.1.7. Cancel [Otkaži].....	13
2.1.5.2. Remove Watch [Ukloni motrenje].....	13
2.1.5.3. Clear Watch [Očisti motrenje]	13
2.1.6. Help [Pomoć]	13
2.2. Oruđna vrpca (engl. <i>toolbar</i> , srp. <i>alatna traka</i>)	13
2.2.1. Step Into [Korakni unutra].....	14
2.2.2. Step Over [Korakni preko]	14
2.2.3. Stop [Zaustavi]	14
2.2.4. Run [Izvršavaj]	14
2.2.5. Restart [Ponovno pokreni]	14

2.2.6. Reload [Ponovno učitaj]	14
2.3. Tipkovnički prečaci	14
2.3.1. F5	14
2.3.2. F11	14
2.3.3. F10	14
2.4. RegisterView [Registarski prikaz]	14
2.4.1. General Purpose [Opća namjena]	14
2.4.1.1. Hexadecimal [Heksadekadski]	14
2.4.1.2. Unsigned Decimal [Nepredznačeni dekadski]	16
2.4.1.3. Signed Decimal [Predznačeni dekadski]	16
2.4.1.4. R0.....	16
2.4.1.5. R1.....	16
2.4.1.6. R2.....	16
2.4.1.7. R3.....	16
2.4.1.8. R4.....	16
2.4.1.9. R5.....	16
2.4.1.10. R6.....	16
2.4.1.11. R7.....	16
2.4.1.12. R8.....	16
2.4.1.13. R9.....	16
2.4.1.14. R10(sl) [R10 (granica stoga)]	16
2.4.1.15. R11(fp) [R11 (pokazivač okvira)]	16
2.4.1.16. R12(ip) [R12 (registar za pozivanje procedure)]	16
2.4.1.17. R13(sp) [R13 (pokazivač stoga)]	16
2.4.1.18. R14(lr) [R14 (povezni registar)]	16
2.4.1.19. R15(pc) [R15 (programsko brojilo)].....	16
2.4.1.20. Negative(N) [Negativno (N)].....	16
2.4.1.21. Zero(Z) [Ništica (Z)]	17
2.4.1.22. Carry(C) [Prijenos (C)]	17
2.4.1.23. Overflow(V) [Preljev (V)]	17
2.4.1.24. IRQ Disable [Prekid onemogućen].....	17
2.4.1.25. FIQ Disable [Brzi prekid onemogućen]	17
2.4.1.26. Thumb(T) [Palčane (T)]	17
2.4.1.27. CPU Mode [Procesorski način rada]	17

2.4.2. Floating Point [Pomična točka]	17
2.5. (naziv učitane datoteke.s)	17
2.5.1. Redci	17
2.5.2. Prvi stupac	17
2.5.3. Drugi stupac (npr. „00001000:EF000203“)	18
2.5.4. Treći stupac (npr. „_START: SWI 0x203“)	18
2.6. MemoryViewx [Memorijski prikaz x]	18
2.6.1. Adresa	18
2.6.2. Strjelica gore	18
2.6.3. Strjelica dolje	18
2.6.4. 8-bit [8-bitno]	18
2.6.4.1. Redci	18
2.6.4.2. Prvi stupac	18
2.6.4.3. Drugi stupac	18
2.6.4.4. Treći stupac	18
2.6.5. 16-bit [16-bitno]	18
2.6.5.1. Redci	18
2.6.5.2. Prvi stupac	19
2.6.5.3. Drugi stupac	19
2.6.6. 32-bit [32-bitno]	19
2.6.6.1. Redci	19
2.6.6.2. Prvi stupac	19
2.6.6.3. Drugi stupac	19
2.7. OutputView [Izlazni prikaz]	19
2.7.1. Console [Konzola]	19
2.7.2. Stdin/Stdout/Stderr [Standardni ulaz/Standardni izlaz/Standardna pogreška]	19
2.8. WatchView [Motrenjski prikaz]	19
2.8.1. Label [Labela]	19
2.8.2. Value [Vrijednost]	20
2.9. StackView [Stožni prikaz]	20
3. Opis naredaba priključka za U/I	20
3.1. SWI 0x00	20
3.2. SWI 0x02	20
3.3. SWI 0x11	20

3.4. SWI 0x12	20
3.5. SWI 0x13	20
3.6. SWI 0x66	20
3.7. SWI 0x68	20
3.8. SWI 0x69	20
3.9. SWI 0x6A	20
3.10. SWI 0x6B	21
3.11. SWI 0x6C	21
3.12. SWI 0x6D	21
4. Pseudonaredbe koje se spominju u priručniku na engleskom i primjerima	21
4.1. .ALIGN	21
4.2. .ASCII	21
4.3. .ASCIZ	21
4.4. .DATA	21
4.5. .END	21
4.6. .EQU	21
4.7. .EXTERN	21
4.8. .GLOBAL	21
4.9. .SKIP	21
4.10. .TEXT	21
4.11. .WORD	22
5. Naredbe koje se spominju u priručniku na engleskom i primjerima	22
5.1. ADD	22
5.2. B	22
5.3. BL	22
5.4. CMP	22
5.5. LDR	22
5.6. LDM	22
5.7. MOV	22
5.8. MUL	22
5.9. RSB	22
5.10. STM	22
5.11. STR	22
5.12. SUB	22

5.13. SWI.....	22
5.14. TST	23
5.15. ORR.....	23
6. Primjeri	23
6.1. Pregled nekih načina naslovljavanja.....	23
6.2. Čitanje sa stdina i ispis na zaslon.....	24
6.3. Otvaranje datoteke i čitanje iz nje.....	26
6.4. Blok <i>if-then-else</i>	27
6.5. Petlja <i>for</i>	28
6.6. Petlja <i>while</i>	28
7. Pisanje koda.....	29
8. Zadatci	29

1. Opis zipa

1.1. a.s

Datoteka u koju možete pisati izvorni kod (vidi 7.).

1.2. ARM.pdf

Ova datoteka.

1.3. ARMPluginInterfaces.dll

1.4. ARMSim.exe

Simulator ARM-a.

1.5. ARMSim.exe.config

1.6. ARMSim.Plugins.EmbestBoardPlugin.dll

1.7. ARMSim.Plugins.UIControls.dll

1.8. ARMSimSamplePlugins.vsi

1.9. ARMSimWindowManager.dll

1.10. ARMSim_UserGuide4Plus.pdf

Priručnik na engleskom jeziku.

1.11. DockingWindows.dll

1.12. DotNetMagic2005.dll

1.13. Primjer 1. Pregled nekih načina naslovljavanja.s

Vidi 6.1.

1.14. Primjer 2. Čitanje sa stdina i ispis na zaslon.s

Vidi 6.2.

1.15. Primjer 3. Otvaranje datoteke i čitanje iz nje.s

Vidi 6.3.

1.16. Primjer 4. Blok if-then-else.s

Vidi 6.4.

1.17. Primjer 5. Petlja for.s

Vidi 6.5.

1.18. Primjer 6. Petlja while.s

Vidi 6.6.

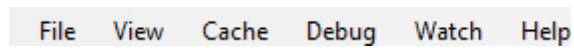
1.19. StaticWindows.dll

1.20. stdin.txt

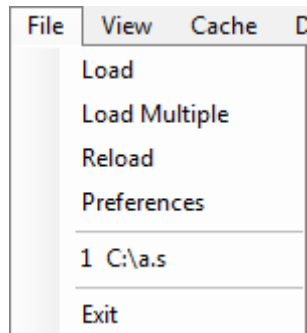
Datoteka koja glumi standardni ulaz (vidi 8.).

2. Opis simulatora

2.1. Glavni izbornik

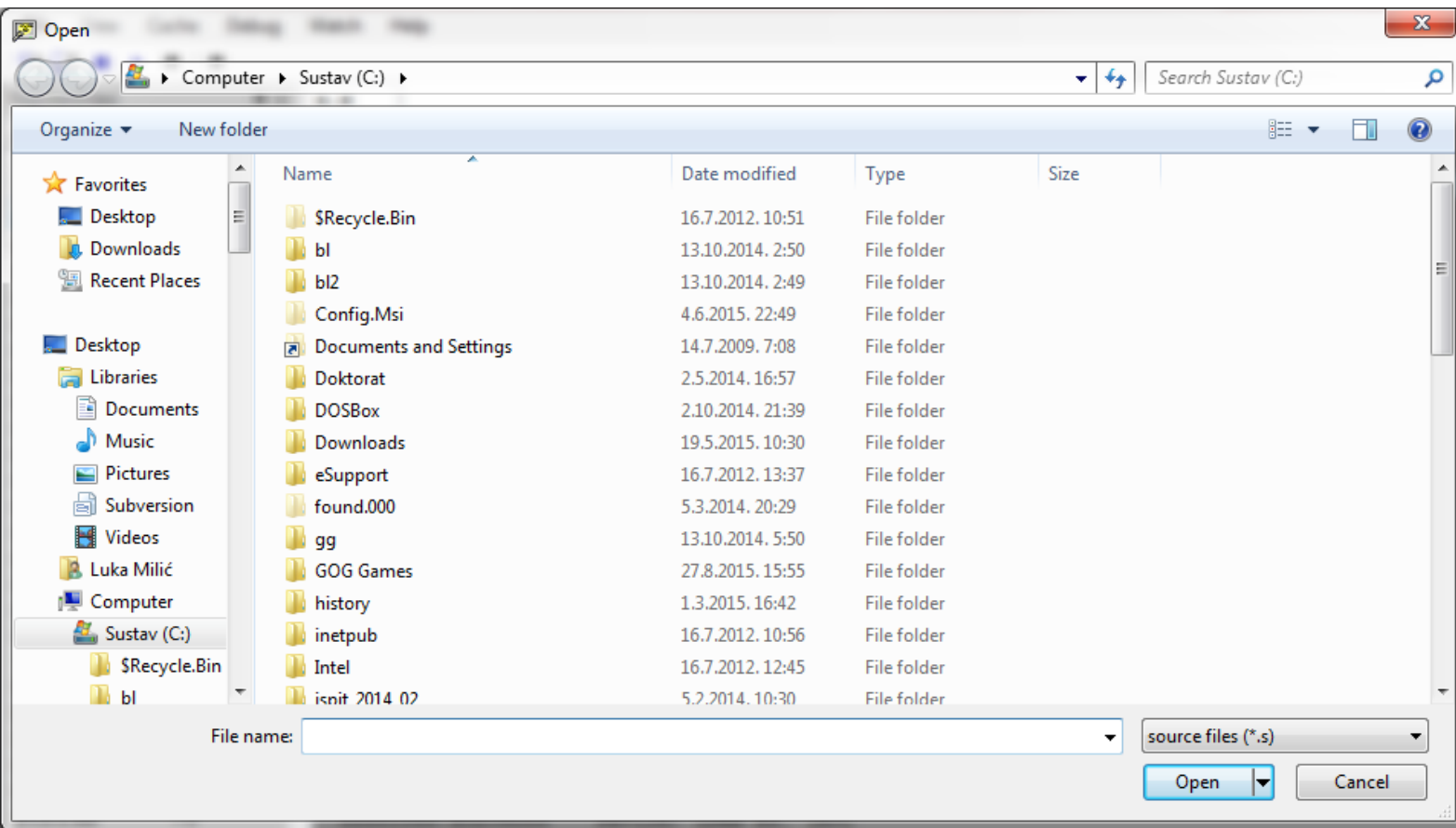


2.1.1. File [Datoteka]



2.1.1.1. Load [Učitaj]

2.1.1.1.1. Standardni okvir za otvaranje datoteka



Odaberite datoteku koju želite otvoriti. Kad se datoteka otvori, odmah se prevede i pokrene.

2.1.1.2. Load Multiple [Učitaj više]

2.1.1.3. Reload [Ponovno učitaj]

Ponovno učitava već učitane datoteke.

2.1.1.4. Preferences [Preferencije]

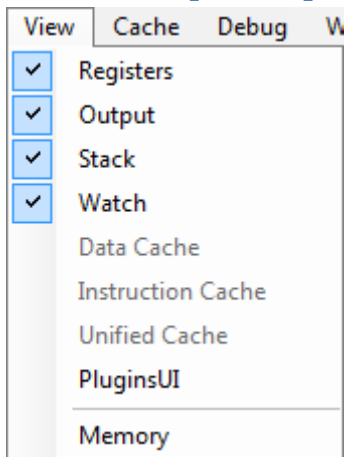
2.1.1.5. (do 4 posljednje otvarane datoteke)

Učitava odbranu datoteku.

2.1.1.6. Exit [Izidi]

Izlazi iz simulatora.

2.1.2. View [Prikaz]



2.1.2.1. Registers [Registri]

Uključuje/isključuje RegisterView (vidi 2.4.).

2.1.2.2. Output [Izlaz]

Uključuje/isključuje OutputView (vidi 2.7.).

2.1.2.3. Stack [Stog]

Uključuje/isključuje StackView (vidi 2.9.).

2.1.2.4 Watch [Motrenje]

Uključuje/isključuje WatchView (vidi 2.8.).

2.1.2.5. Data Cache [Podatkovna priručna memorija]

2.1.2.6. Instruction Cache [Naredbena priručna memorija]

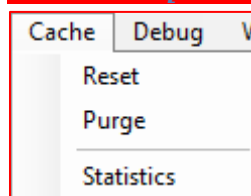
2.1.2.7. Unified Cache [Ujedinjena priručna memorija]

2.1.2.8. PluginsUI [Korisničko sučelje priključaka]

2.1.2.9. Memory [Memorija]

Uključuje dodatan MemoryView (vidi 2.6.).

2.1.3. Cache [Priručna memorija]



2.1.4. Debug [Uklanjanje pogrešaka]

Debug	Watch	Help
Run		F5
Step Into		F11
Step Over		F10
Restart		
Stop		
Clear All Breakpoints		

2.1.4.1. Run [Izvršavaj]

Izvršava program.

2.1.4.2. Step Into [Korakni unutra]

Izvršava sljedeću naredbu u programu. Ako je sljedeća naredba poziv funkcije, ulazi u funkciju.

2.1.4.3. Step Over [Korakni preko]

Izvršava sljedeću naredbu u programu. Ako je sljedeća naredba poziv funkcije, izvršava cijelu funkciju.

2.1.4.4. Restart [Ponovno pokreni]

Ponovno pokreće program.

2.1.4.5. Stop [Zaustavi]

Zaustavlja izvršavanje programa.

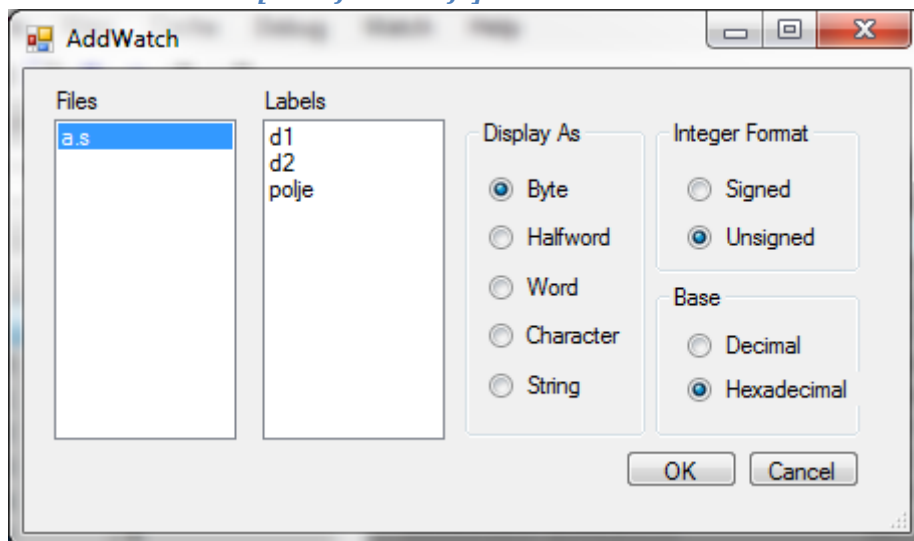
2.1.4.6. Clear All Breakpoints [Očisti sve prekidne točke]

Uklanja sve prekidne točke.

2.1.5. Watch [Motrenje]

Watch	Help
Add Watch	
Remove Watch	
Clear All	

2.1.5.1. Add Watch [Dodaj motrenje]



2.1.5.1.1. Files [Datoteke]

2.1.5.1.2. Labels [Labele]

Labela memorijskoga mjesta koje želite pratiti.

2.1.5.1.3. Display As [Prikaži kao]

Način prikaza memorijskoga mjesta.

2.1.5.1.4. Integer Format [Cjelobrojno obličje]

Način prikaza memorijskoga mjesta kao cijeloga broja.

2.1.5.1.5. Base [Baza]

Baza cijeloga broja.

2.1.5.1.6. OK [U redu]

Postavlja odbranu labelu u WatchView (vidi 2.8.).

2.1.5.1.7. Cancel [Otkazi]

Poništava izbor.

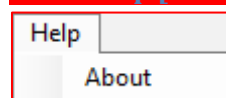
2.1.5.2. Remove Watch [Ukloni motrenje]

Uklanja motrenje odbrano u WatchViewu (vidi 2.8.).

2.1.5.3. Clear Watch [Očisti motrenje]

Uklanja sva motrenja.

2.1.6. Help [Pomoć]



2.2. Oruđna vrpca (engl. toolbar, srp. alatna traka)



2.2.1. Step Into [Korakni unutra]

Vidi 2.4.2.

2.2.2. Step Over [Korakni preko]

Vidi 2.4.3.

2.2.3. Stop [Zaustavi]

Vidi 2.4.5.

2.2.4. Run [Izvršavaj]

Vidi 2.4.1.

2.2.5. Restart [Ponovno pokreni]

Vidi 2.4.4.

2.2.6. Reload [Ponovno učitaj]

Vidi 2.1.3.

2.3. Tipkovnički prečaci

2.3.1. *F5*

Vidi 2.4.1.

2.3.2. *F11*

Vidi 2.4.2.

2.3.3. *F10*

Vidi 2.4.3.

2.4. RegisterView [Registarski prikaz]

Crvena će slova u programu označivati da se je na to registarsko mjesto pisalo od zadnje stanke u izvođenju koda.

2.4.1. General Purpose [Opća namjena]

2.4.1.1. *Hexadecimal* [Heksadekadski]

Prikazuje vrijednosti registara kao heksadekadske brojeve.

RegistersView	RegistersView	RegistersView
General Purpose Floating Point	General Purpose Floating Point	General Purpose Floating Point
Hexadecimal	Hexadecimal	Hexadecimal
Unsigned Decimal	Unsigned Decimal	Unsigned Decimal
Signed Decimal	Signed Decimal	Signed Decimal
R0 : 00000000	R0 : 0	R0 : 0
R1 : 00000000	R1 : 0	R1 : 0
R2 : 00000000	R2 : 0	R2 : 0
R3 : 00000000	R3 : 0	R3 : 0
R4 : 00000000	R4 : 0	R4 : 0
R5 : 00000000	R5 : 0	R5 : 0
R6 : 00000000	R6 : 0	R6 : 0
R7 : 00000000	R7 : 0	R7 : 0
R8 : 00000000	R8 : 0	R8 : 0
R9 : 00000000	R9 : 0	R9 : 0
R10 (s1) : 00000000	R10 (s1) : 0	R10 (s1) : 0
R11 (fp) : 00000000	R11 (fp) : 0	R11 (fp) : 0
R12 (ip) : 00000000	R12 (ip) : 0	R12 (ip) : 0
R13 (sp) : 00005400	R13 (sp) : 21504	R13 (sp) : 21504
R14 (lr) : 00000000	R14 (lr) : 0	R14 (lr) : 0
R15 (pc) : 00001000	R15 (pc) : 4096	R15 (pc) : 4096
-----	-----	-----
CPSR Register	CPSR Register	CPSR Register
Negative (N) : 0	Negative (N) : 0	Negative (N) : 0
Zero (Z) : 0	Zero (Z) : 0	Zero (Z) : 0
Carry (C) : 0	Carry (C) : 0	Carry (C) : 0
Overflow (V) : 0	Overflow (V) : 0	Overflow (V) : 0
IRQ Disable : 1	IRQ Disable : 1	IRQ Disable : 1
FIQ Disable : 1	FIQ Disable : 1	FIQ Disable : 1
Thumb (T) : 0	Thumb (T) : 0	Thumb (T) : 0
CPU Mode : System	CPU Mode : System	CPU Mode : System
-----	-----	-----
0x000000df	0x000000df	0x000000df

2.4.1.2. Unsigned Decimal [Nepredznačeni dekadski]

Prikazuje vrijednosti registara kao nepredznačene dekadске brojeve.

2.4.1.3. Signed Decimal [Predznačeni dekadski]

Prikazuje vrijednosti registara kao predznačene dekadске brojeve.

2.4.1.4. R0

Vrijednost registra R0.

2.4.1.5. R1

Vrijednost registra R1.

2.4.1.6. R2

Vrijednost registra R2.

2.4.1.7. R3

Vrijednost registra R3.

2.4.1.8. R4

Vrijednost registra R4.

2.4.1.9. R5

Vrijednost registra R5.

2.4.1.10. R6

Vrijednost registra R6.

2.4.1.11. R7

Vrijednost registra R7.

2.4.1.12. R8

Vrijednost registra R8.

2.4.1.13. R9

Vrijednost registra R9.

2.4.1.14. R10(sl) [R10 (granica stoga)]

2.4.1.15. R11(fp) [R11 (pokazivač okvira)]

2.4.1.16. R12(ip) [R12 (registar za pozivanje procedure)]

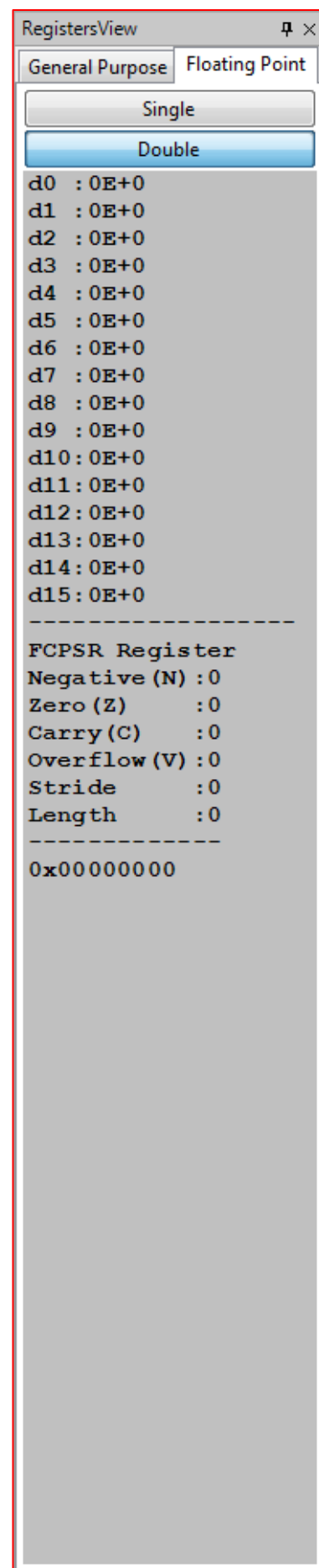
2.4.1.17. R13(sp) [R13 (pokazivač stoga)]

2.4.1.18. R14(lr) [R14 (povezni registar)]

2.4.1.19. R15(pc) [R15 (programsko brojilo)]

Vrijednost registra R15 (programskoga brojila).

2.4.1.20. Negative(N) [Negativno (N)]



Vrijednost zastavice negativno (N).

2.4.1.21. Zero(Z) [Ništica (Z)]

Vrijednost zastavice ništica (Z).

2.4.1.22. Carry(C) [Prijenos (C)]

Vrijednost zastavice prijenosa (C).

2.4.1.23. Overflow(V) [Preljev (V)]

Vrijednost zastavice preljeva (V).

2.4.1.24. IRQ Disable [Prekid onemogućen]

2.4.1.25. FIQ Disable [Brzi prekid onemogućen]

2.4.1.26. Thumb(T) [Palčane (T)]

2.4.1.27. CPU Mode [Procesorski način rada]

2.4.2. Floating Point [Pomična točka]

2.5. (naziv učitane datoteke.s)

Obojeni redak u programu označuje da se dotična naredba sljedeća treba izvoditi.

```
a.s
        .GLOBAL _START
        .TEXT
00001000:EF000203    _START: SWI 0x203
00001004:E3500000    CMP R0, #0
00001008:0AFFFFFC    BEQ _START
0000100C:E3500001    CMP R0, #1
00001010:059F00B8    LDREQ R0, =D1
00001014:159F00B8    LDRNE R0, =D2
00001018:E3A01000    MOV R1, #0
0000101C:EF000066    SWI 0x66
00001020:E59F10B0    LDR R1, =POLJE
00001024:E3A02050    MOV R2, #80
00001028:E1A03000    MOV R3, R0
0000102C:EF00006A    SWI 0x6A
00001030:E1A00003    MOV R0, R3
00001034:EF000068    SWI 0x68
00001038:E1A02001    MOV R2, R1
0000103C:E3A00000    MOV R0, #0
00001040:E3A01000    MOV R1, #0
00001044:EF000204    SWI 0x204
00001048:E3A01001    MOV R1, #1
0000104C:E2822028    ADD R2, R2, #40
00001050:EF000204    SWI 0x204
00001054:E2422028    SUB R2, R2, #40
00001058:E3A01003    MOV R1, #3
0000105C:E1A03002    MOV R3, R2
00001060:E3A04001    MOV R4, #1
00001064:E3A05000    MOV R5, #0
00001068:E5D36000    PETLJA: LDRB R6, [R3]
0000106C:E2833001    ADD R3, R3, #1
00001070:E356002E    CMP R6, #46
00001074:0A00000D    BEQ TOCKA
00001078:E3560020    CMP R6, #32
0000107C:0A000002    BEQ RAZMAK
00001080:E356005B    CMP R6, #91
00001084:B3A04000    MOVLT R4, #0
00001088:EAF00006    BAL PETLJA
0000108C:E3540000    RAZMAK: CMP R4, #0
00001090:12855001    ADDNE R5, R5, #1
00001094:05434001    STREQB R4, [R3, #-1]
```

2.5.1. Redci

Strojne naredbe.

2.5.2. Prvi stupac

Postoji li prekidna točka u retku. Crljena će točka označivati da postoji. Dva puta pritisnite za uključivanje/isključivanje prekidne točke.

2.5.3. Drugi stupac (npr. „00001000:EF000203“)

Heksadekadski prikaz adrese memorijskoga mjesta u retku, dvotočje, heksadekadski prikaz sadržaja memorijskoga mjesta u retku.

2.5.4. Treći stupac (npr. „_START: SWI 0x203“)

Odgovarajući redak zbirnoga koda.

2.6. MemoryViewx [Memorijski prikaz x]

2.6.1. Adresa

Heksadekadski prikaz adrese memorijskoga mjesta od kojega se memorija prati.

2.6.2. Strjelica gore

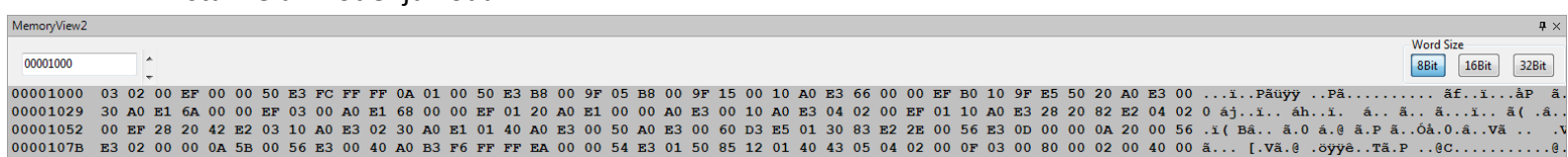
Pomicanje adrese za jednu stranicu gore.

2.6.3. Strjelica dolje

Pomicanje adrese za jednu stranicu dolje.

2.6.4. 8-bit [8-bitno]

Crvena će slova u programu označivati da se je na ta memorijska mjesta pisalo od zadnje stanke u izvođenju koda.



2.6.4.1. Redci

Blokovi memorije.

2.6.4.2. Prvi stupac

Heksadekadski prikaz prvoga memorijskoga mjesta u retku.

2.6.4.3. Drugi stupac

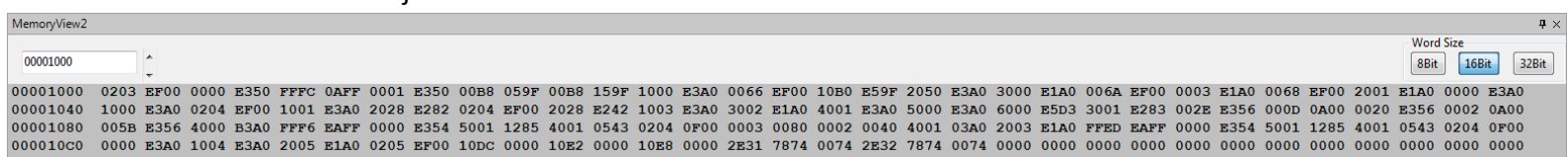
Sadržaj memorijskoga mjesta bajt po bajt kao heksadekadski broj, odvojeno razmakom.

2.6.4.4. Treći stupac

Sadržaj memorijskoga mjesta bajt po bajt kao ASCII-jev znak.

2.6.5. 16-bit [16-bitno]

Crvena će slova u programu označivati da se je na ta memorijska mjesta pisalo od zadnje stanke u izvođenju koda.



2.6.5.1. Redci

Blokovi memorije.

2.6.5.2. Prvi stupac

Heksadekadska adresa memorijskoga mjesta u retku.

2.6.5.3. Drugi stupac

Sadržaj memorijskoga mjesta dva bajta po dva bajta kao heksadekadski broj, odvojeno razmakom.

2.6.6. 32-bit [32-bitno]

Crvena će slova u programu označivati da se je na ta memorijska mjesta pisalo od zadnje stanke u izvođenju koda.

MemoryView2																Word Size		
<div>00001000</div>																8Bit	16Bit	32Bit
00001000	EF000203	E3500000	0AFFFFFF	E3500001	059F00B8	159F00B8	E3A01000	EF000066	E59F10B0	E3A02050	E1A03000	EF00006A	E1A00003	EF000068	E1A02001	E3A00000		
00001040	E3A01000	EF000204	E3A01001	E2822028	EF000204	E2422028	E3A01003	E1A03002	E3A04001	E3A05000	E5D36000	E2833001	E356002E	0A00000D	E3560020	0A000002		
00001080	E356005B	B3A04000	EAF00000	E3540000	12855001	05434001	0F000204	00800003	00400002	03A04001	E1A02003	EAF00000	E3540000	12855001	05434001	0F000204		
000010C0	E3A00000	E3A01004	E1A02005	EF000205	000010DC	000010E2	000010E8	78742E31	2E320074	00747874	00000000	00000000	00000000	00000000	00000000	00000000		

2.6.6.1. Redci

Blokovi memorije.

2.6.6.2. Prvi stupac

Heksadekadska adresa memorijskoga mjesta u retku.

2.6.6.3. Drugi stupac

Sadržaj memorijskoga mjesta četiri bajta po četiri bajta kao heksadekadski broj, odvojeno razmakom.

2.7. OutputView [Izlazni prikaz]

2.7.1. Console [Konzola]

OutputView

Console

Stdin/Stdout/Stderr

Loading assembly language file C:\a.s

OutputView

WatchView

Ispisuje se izlaz simulatora.

2.7.2. Stdin/Stdout/Stderr [Standardni ulaz/Standardni izlaz/Standardna pogriješka]

OutputView

Console

Stdin/Stdout/Stderr

OutputView

WatchView

Standardni ulaz, standardni izlaz i standardni izlaz za pogriješke pokrenutoga programa.

2.8. WatchView [Motrenjski prikaz]

WatchView

Label	Value

OutputView

WatchView

2.8.1. Label [Labela]

Labela memorijskoga mjesta koje se prati.

2.8.2. Value [Vrijednost]

Vrijednost memorijskoga mjesta koje se prati.

2.9. StackView [Stožni prikaz]

3. Opis naredaba priključka za U/I

3.1. SWI 0x00

Piši znak na standardni izlaz. U registru R0 treba biti znak.

3.2. SWI 0x02

Piši znakovni niz na standardni izlaz. U registru R0 treba biti dotični ASCII-jev znakovni niz završen ničicom.

3.3. SWI 0x11

Završi program.

3.4. SWI 0x12

3.5. SWI 0x13

3.6. SWI 0x66

Otvori datoteku. U registru R0 treba biti ASCII-jev znakovni niz završen ničicom u kojem piše datotekin naziv. U registru R1 treba biti 0 ako se iz datoteke čita, 1 ako se u datoteku piše i 2 ako se u datoteku dodaje. U registru će R0 poslije biti datotečna ručica ako se datoteka uspješno otvori, -1 inače.

3.7. SWI 0x68

Zatvori datoteku. U registru R0 treba biti dotična datotečna ručica.

3.8. SWI 0x69

Piši znakovni niz u datoteku ili na standardni izlaz. U registru R0 treba biti datotečna ručica ako se piše u datoteku, 1 ako se piše na standardni izlaz. U registru R1 treba biti dotični ASCII-jev znakovni niz završen ničicom.

3.9. SWI 0x6A

Čitaj znakovni niz iz datoteke. U registru R0 treba biti dotična datotečna ručica. U registru R1 treba biti memorijsko mjesto kamo će se staviti pročitani niz. U registru R2 treba biti najveća moguća njegova veličina. U registru će R0 poslije biti broj nepročitanih bajtova.

StackView	
00005398:	81818181
0000539C:	81818181
000053A0:	81818181
000053A4:	81818181
000053A8:	81818181
000053AC:	81818181
000053B0:	81818181
000053B4:	81818181
000053B8:	81818181
000053BC:	81818181
000053C0:	81818181
000053C4:	81818181
000053C8:	81818181
000053CC:	81818181
000053D0:	81818181
000053D4:	81818181
000053D8:	81818181
000053DC:	81818181
000053E0:	81818181
000053E4:	81818181
000053E8:	81818181
000053EC:	81818181
000053F0:	81818181
000053F4:	81818181
000053F8:	81818181
000053FC:	81818181
00005400:	81818181
00005404:	81818181
00005408:	81818181
0000540C:	81818181
00005410:	81818181
00005414:	81818181
00005418:	81818181
0000541C:	81818181
00005420:	81818181
00005424:	81818181
00005428:	81818181
0000542C:	81818181
00005430:	81818181
00005434:	81818181
00005438:	81818181
0000543C:	81818181
00005440:	81818181
00005444:	81818181
00005448:	81818181
0000544C:	81818181
00005450:	81818181
00005454:	81818181
00005458:	81818181
0000545C:	81818181
00005460:	81818181
00005464:	81818181
00005468:	81818181

3.10. SWI 0x6B

Piši cijeli broj u datoteku ili na standardni izlaz. U registru R0 treba biti datotečna ručica ako se piše u datoteku, 1 ako se piše na standardni izlaz. U registru R1 treba biti dotični cijeli broj.

3.11. SWI 0x6C

Čitaj cijeli broj iz datoteke. U registru R0 treba biti dotična datotečna ručica. U registru R1 će R0 poslije biti pročitani cijeli broj.

3.12. SWI 0x6D

4. Pseudonaredbe koje se spominju u priručniku na engleskom i primjerima

Neke će vam od ovih trebati za rješavanje vašega zadatka. Skladnju (sintaksu) potražite na Googleu. Radi se u GNU-ovim zbirnim pseudonaredbama (engl. *GNU assembler directives*).

4.1. .ALIGN

engl. *align* -> poravnava adresu koda

4.2. .ASCII

engl. *ASCII* -> postavlja ASCII-jev znakovni niz u kod

4.3. .ASCIZ

engl. *ASCII zero* -> postavlja ASCII-jev znakovni niz završen ničicom na adresu u kodu

4.4. .DATA

engl. *data* -> određuje da je sljedeći dio koda i za čitanje i za pisanje

4.5. .END

engl. *end* -> određuje završetak koda

4.6. .EQU

engl. *equal* -> određuje makroe

4.7. .EXTERN

engl. *external* -> određuje eksterne labele

4.8. .GLOBAL

engl. *global* -> određuje globalne labele

4.9. .SKIP

engl. *skip* -> alokira dio memorije

4.10. .TEXT

engl. *text* -> određuje da je sljedeći dio koda samo za čitanje

4.11. .WORD

engl. *word* -> postavlja niz riječi u kod

5. Naredbe koje se spominju u priručniku na engleskom i primjerima

Neke će vam od ovih trebati za rješavanje vašeg zadatka. Skladnju (sintaksu) potražite na Googleu. Radi se o ARM-ovim zbirnim naredbama (engl. *ARM assembler instructions*).

5.1. ADD

engl. *add* -> zbrajanje

5.2. B

engl. *branch* -> skakanje

5.3. BL

engl. *branch link* -> skakanje u funkciju

5.4. CMP

engl. *compare* -> usporedba

5.5. LDR

engl. *load register* -> učitavanje

5.6. LDM

engl. *load multiple* -> višestruko učitavanje

5.7. MOV

engl. *move* -> preslikavanje

5.8. MUL

engl. *multiply* -> množenje

5.9. RSB

engl. *reverse subtract* -> obrnuto oduzimanje

5.10. STM

engl. *store multiple* -> višestruko spremanje

5.11. STR

engl. *store register* -> spremanje

5.12. SUB

engl. *subtract* -> oduzimanje

5.13. SWI

engl. *software interrupt* -> programski prekid

5.14. TST

engl. *test* -> provjera je li

5.15. ORR

engl. *or* -> logičko ili

6. Primjeri

6.1. Pregled nekih načina naslovljavanja

@početak

.GLOBAL _START

.TEXT

_START: @izravno naslovljavanje
LDR R0, =0x10002000

 @izravno naslovljavanje
MOV R0, #10

 @inicijalizacija
LDR R0, =0xFFFFFFFF
LDR R1, =0x01234567

 @izravno naslovljavanje registara
MOV R0, R1

 @inicijalizacija
LDR R0, =0xFFFFFFFF
LDR R1, =0x00001000

 @spremanje poluriječi
STRH R0, [R1]

 @inicijalizacija
LDR R0, =0xFFFFFFFF
LDR R1, =0x00001000

 @spremanje bajta
STRB R0, [R1]

 @inicijalizacija

```
LDR R0, =0x0000FFFF
LDR R1, =0x00000000
```

```
@neizravno naslovljavanje s pomoću registra
LDR R0, [R1]
```

```
@inicijalizacija
LDR R0, =0xFFFFFFFF
LDR R1, =0x00001000
```

```
@neizravno naslovljavanje s pomoću registra s pomakom poslije
LDR R0, [R1], #4
```

```
@inicijalizacija
LDR R0, =0xFFFFFFFF
LDR R1, =0x00001000
```

```
@neizravno naslovljavanje s pomoću registra s pomakom
LDR R0, [R1, #4]
```

```
@inicijalizacija
LDR R0, =0xFFFFFFFF
LDR R1, =0x00001000
```

```
@neizravno naslovljavanje s pomoću registra s pomakom prije
LDR R0, [R1, #4]!
```

```
@inicijalizacija
LDR R0, =0xFFFFFFFF
LDR R1, =0x00001000
LDR R2, =0x00001000
```

```
@neizravno naslovljavanje s pomoću registra s bačvastim sklopom za posmak
LDR R0, [R1, R2, LSL #2]
```

```
@kraj rada simulatora
SWI 0x11
```

```
.END
```

6.2. Čitanje sa stdina i ispis na zaslon

```
.GLOBAL _START
```



```

.TEXT
_START:  @postavlja adresu naziva datoteke u R0
        LDR R0, =STDIN
        @postavlja način otvaranja datoteke u R1
        MOV R1, #0
        @otvara datoteku
        SWI 0x66
        @sprema datotečnu ručicu
        MOV R9, R0
        @čita cijeli broj iz datoteke
        SWI 0x6C

        @postavlja cijeli broj u R1
        MOV R1, R0
        @postavlja datotečnu ručicu u R0
        MOV R0, #1
        @piše cijeli broj na standardni izlaz
        SWI 0x6B

        @učitava znakovni niz i sprema ga u radnu memoriju na mjesto SPREMNIK
        @postavlja datotečnu ručicu u R0
        MOV R0, R9
        @postavlja adresu kamo spremiti u R1
        LDR R1, =SPREMNIK
        @postavlja maksimalnu veličinu znakovnoga niza u R2
        MOV R2, #80
        @čita znakovni niz iz datoteke
        SWI 0x6A
        @postavlja datotečnu ručicu u R0
        MOV R0, R9
        @zatvara datoteku
        SWI 0x68

        @postavlja datotečnu ručicu u R0
        MOV R0, #1
        @piše znakovni niz na standardni izlaz
        SWI 0x69

        @postavlja adresu znakovnoga niza u R1
        LDR R1, =POZDRAV
        @piše znakovni niz na standardni izlaz

```

SWI 0x69

@zaustavlja simulator

SWI 0x11

@stdin

STDIN: .ASCIZ "stdin.txt"

@spremnik za ulazni tekst

SPREMNIK: .SKIP 80

@tekst koji završava ničicom

POZDRAV: .ASCIZ "Pozdrav svima!"

.END

6.3. Otvaranje datoteke i čitanje iz nje

.GLOBAL _START

.TEXT

_START: @otvaranje stdina

LDR R0, =STDIN

MOV R1, #0

SWI 0x66

MOV R9, R0

@čitanje naziva datoteke

LDR R1, =UNOS1

MOV R2, #80

SWI 0x6A

MOV R0, R9

SWI 0x68

@otvaranje datoteke

LDR R0, =UNOS1

MOV R1, #0

SWI 0x66

@čitanje iz datoteke

LDR R1, =UNOS2

MOV R2, #80

SWI 0x6A

@ispis teksta

MOV R0, #1

SWI 0x69

@zaustavlja simulator

SWI 0x11

```

        .DATA
STDIN:   .ASCIZ "stdin.txt"
UNOS1:   .SKIP 80
UNOS2:   .SKIP 80
        .END

```

6.4. Blok *if-then-else*

```

        .GLOBAL _START
        .TEXT

_START:  LDR R0, =STDIN
        MOV R1, #0
        SWI 0x66
        MOV R9, R0

        SWI 0x6C
        MOV R1, R0
        MOV R0, R9
        SWI 0x68

        CMP R1, #2
        BLT X1

        MOV R0, #1
        LDR R1, =DOBRO
        SWI 0x69
        BAL X2

X1:      MOV R0, #1
        LDR R1, =SLABO
        SWI 0x69

X2:      SWI 0x11

        .DATA
STDIN:   .ASCIZ "stdin.txt"
DOBRO:   .ASCIZ "SUPER "
SLABO:   .ASCIZ "SLABO "

        .END

```

6.5. Petlja *for*

```
.GLOBAL _START
.TEXT

_START:    LDR R0, =STDIN
           MOV R1, #0
           SWI 0x66
           MOV R9, R0

           SWI 0x6C
           MOV R2, R0

           MOV R0, R9
           SWI 0x6C
           MOV R1, R0

           MOV R0, R9
           SWI 0x68

X2:        CMP R1, R2
           BGT X1
           MOV R0, #1
           SWI 0x6B
           ADD R1, R1, #1
           BAL X2

X1:        SWI 0x11

           .DATA
STDIN:     .ASCIZ "stdin.txt"

           .END
```

6.6. Petlja *while*

```
.GLOBAL _START
.TEXT

_START:    LDR R0, =STDIN
           MOV R1, #0
           SWI 0x66
           MOV R9, R0
```

```

        SWI 0x6C
        MOV R2, R0
        MOV R0, R9
        SWI 0x68
        BAL X2

X1:      MOV R0, #1
        LDR R1, =ISPIS
        SWI 0x69
        SUB R2, R2, #1

X2:      CMP R2, #0
        BHI X1

        SWI 0x11

        .DATA
STDIN:   .ASCIZ "stdin.txt"
ISPIS:   .ASCIZ "cool broj "

        .END

```

7. Pisanje koda

Kod pišete u kojem bilo tekstnom uređivaču. Treba paziti da je datotečna proširka .s.

Za razliku od uobičajenoga, kôd se u GNU-ovu zbirnom prevodiocu obično piše malim slovima i u jednom stupcu. Toga se u tekstu nije držalo, nego se piše velikim slovima i u dvama stupcima.

Kod pisanja ARM-ova kôda zgodno je uočiti da više-manje svaka naredba može biti uvjetna, kao i da ne mora postavljati zastavice. To možete potražiti na Googleu, ovdje nema primjera.

8. Zadatci

U sljedećim zadatcima, kad vam bude pisalo „učitajte sa standardnoga ulaza“, računajte da piše „učitajte iz datoteke stdin.txt“. Zadatci su isti kao i oni za Motorolu. Zadatci nisu isti kao i lani!

Zadaci za studente

Studenti trebaju riješiti zadatak pod brojem koji se dobije na slijedeći način: Zadnji broj matičnog broja studenta (ne broj godine) određuje broj zadatka.

Npr. Matični broj studenta je 32214 / 11. **Zadatak broj 4**

Napomena:

- Zabranjeno je koristiti izraze if, for, while, repeat kao što je to prikazano u Help->Assembler Commands->StructuredControl. Koristite osnovne naredbe kao što je to prikazano u primjerima 4., 5. i 6.
- Prilikom predaje zadataka potrebno je za svaku instrukciju programa znati što radi i koje resurse (memorijske lokacije ili registre) koristi.
- Program koji se predaje ne smije imati komentare.
- Rok za predaju programa je zadnji tjedan izvođenja nastave u zimskom semestru.
- Program je moguće predati, prezentirati i dati odgovore na pitanja isključivo u terminima seminarskih vježbi.
- Prije predaje programa potrebno je program pohraniti na stranice kolegija.

Sustav bodovanja:

- Sustav bodovanja će biti objavljen na webu.
- Neuspješna predaja zadatka tj. nerazumijevanje zadatka koji se predaje povlači oduzimanje 5 bodova od trenutno ukupno mogućih bodova.
- Uzastopno neuspješno predavanje zadatka povlači promjenu zadatka.

Zadaci

0. Neka se u datoteci nalazi niz pozitivnih brojeva zapisani u ASCII formatu i odvojenih razmakom. Neka se brojevi kreću u rasponu od 0 do 999.
 - Učitajte naziv datoteke preko tipkovnice.
 - Ispišite sadržaj datoteke na ekran.
 - Ispišite sve brojeve iz datoteke koji su veći od 50 i manji od 300.
 - Ispišite aritmetičku sredinu brojeva koji su veći od 50 i manji od 300.

Primjer teksta u datoteci (tekst neka završi sa točkom):

56 789 200 2 97 635.

Rješenje:

56 789 200 2 98 635

Brojevi veći od 50 i manji od 300:

56 200 98

Aritmetička sredina: 118

1. Neka datoteka sadrži proizvoljni broj riječi (mala slova abecede i brojevi zapisani u ASCII formatu).
 - Učitajte naziv datoteke preko tipkovnice.
 - Ispišite sadržaj datoteke na ekran.
 - Ispišite sve riječi iz datoteke sortirano. (Pretpostavlja se da će se učitati samo numerički znakovi i slova engleske abecede.).

Primjer teksta u datoteci (tekst neka završi sa točkom):

volim rjesavati zadatke iz arhitekture.

*Rješenje:
arhitekture
iz
rjesavati
volim
zadatke*

2. Neka se u datoteci nalazi niz riječi koji završava s točkom.
 - Učitajte naziv datoteke preko tipkovnice.
 - Ispišite tekst iz datoteke na ekran.
 - Odredite ukupni broj suglasnika.
 - Ispišite riječi koje imaju više od tri samoglasnika.

Primjer teksta u datoteci (tekst neka završi sa točkom):

Volim rjesavati zadatke iz arhitekture.

*Rješenje:
Ukupni broj suglasnika: 19
Rijeci s vise od dva samoglasnika: rjesavati zadatke arhitekture*

3. Neka datoteka sadrži brojeve različite duljine u rasponu od 0-999 (zapisane u ASCII formatu).
 - Učitajte naziv datoteke preko tipkovnice.
 - Ispišite sadržaj datoteke na ekran.
 - Odredite i ispišite najveći.
 - Ispišite sumu najvećeg i najmanjeg broja.

Primjer teksta u datoteci (zadnji broj ima točku iza sebe):

222 120 300 450 201 117 834 456.

Rješenje:

Najveći broj: 834

Najmanji broj: 117

Suma najvećeg i najmanjeg broja: 951

4. Neka se u datoteci nalazi niz riječi koji završava s točkom.

- Učitajte naziv datoteke preko tipkovnice.
- Ispišite sadržaj datoteke na ekran.
- Ispišite riječi koje se pojavljuju više puta.

Primjer teksta u datoteci (tekst neka završi sa točkom):

Kad letim u snu tad letim u raketi.

Rješenje:

letim

u

5. Neka se u datoteci nalazi niz riječi koji završava s točkom.

- Učitajte naziv datoteke preko tipkovnice.
- Ispišite sadržaj datoteke na ekran.
- Učitajte slovo korištenjem tipkovnice
- Ispišite sve riječi iz datoteke koje sadrže to slovo obrnutim redoslijedom slova.

Primjer teksta u datoteci (tekst neka završi sa točkom):

Ovo je neki tekst u datoteci koji završava s točkom.

Unesite slovo:

t

Rješenje:

tsket

icetotda

mokcot

6. Neka datoteka sadrži brojeve različite duljine u rasponu od 0-999 (zapisane u ASCII formatu).

- Učitajte naziv datoteke preko tipkovnice.
- Ispišite sadržaj datoteke na ekran.
- Ispišite sve jednoznamenaste brojeve.
- Ispišite aritmetičku sredinu dvoznamenkastih brojeva.

Primjer teksta u datoteci (zadnji broj ima točku iza sebe):

222 10 300 7 450 201 9 117 38 456.

Rješenje:

Jednoznamenkasti brojevi: 7 9

Aritmetička sredina dvoznamenkastih brojeva: 24

7. Neka se u datoteci nalazi niz riječi koji završava s točkom.

- Učitajte naziv datoteke preko tipkovnice.
- Ispišite sadržaj datoteke na ekran.
- Pronađite i ispišite riječi koje u sebi imaju samo neparne znamenke.

Primjer teksta u datoteci (tekst neka završi s točkom):

pla7va rib8a r3a5ku gr36ize gric.

Rješenje:

pla7va r3a5ku

8. Neka se u datoteci nalaze brojevi u rasponu od 0 do 999 odvojeni razmakom i zapisani u ASCII formatu.

- Učitajte naziv datoteke preko tipkovnice.
- Ispišite brojeve na ekran
- Ispišite brojeve djeljive s brojem 3

Primjer teksta u datoteci (zadnji broj neka završi sa točkom):

123 46 34 6 345 7.

Rješenje:

Brojevi djeljivi s 3: 123 46 6 345

9. Neka datoteka sadrži proizvoljni broj riječi (mala slova abecede i znamenke zapisani u ASCII formatu).

- Učitajte naziv datoteke preko tipkovnice.
- Ispišite sadržaj datoteke na ekran.
- Ispišite na ekran sve riječi koje u sebi imaju barem dva samoglasnika.

Primjer teksta u datoteci (tekst neka završi sa točkom):

Tražim riječi koje imaju barem tri samoglasnika.

Rješenje:

riječi imaju samoglasnika