Pojasnjenje problema iz zadace 1

Problem segmentation fault-a u zadatku 1 kod nekih kolega:

Naime, znamo da se nakon mapiranja programa u memoriju postavlja programski brojac na entry point, tacku koja oznacava virtualnu adresu na kojoj ce se program poceti izvrsavati. Ulazna tacka se kod ELF datoteka obicno specificira simbolom "_start". Funkcija _start inicijalizira stack i pozove fju __libc_start_main, koja poziva fju main. Ova fja takode poziva exit kada se vratimo iz fje main.
Fja __libc_start_main se ne brine o $s? registrima, odnosno, fja main ne bi trebala mijenjati stanja ovih registara. Mi u nasem zadatku mijenjamo stanja $s registara.
Segmentation fault se desio jer smo promijenili stanje $s0 registra i nakon sto smo se vratili iz fje main u fju __libc_start_main instrukcijom jr $ra, sljedeca instrukcija je
“lw t9, -32000(s0)”, a s obzirom da smo izmijenili registar $s0, pristupamo nevalidnoj adresi.

O cuvanju stanja $t i $s registara cemo detaljnije govoriti kada budemo radili funkcije, i tada ce biti sve jasno.

Mozemo izbjeci ovaj problem ako koristimo registar $s2 umjesto $s0, a $s3 koristimo za ucitavanje adrese varijabli broj1 i broj 2.
Generalno, izbjegavajte koristenje $s registara u ovom trenutku u zadacima 2,3,4, i 5.
Takoder, mozete ignorisati errore koji vam se trenutno javljaju nakon instrukcije jr $ra u fji main. Ako je stanje registara i memorije prije te instrukcije uredu, onda je vas program ispravan (u ovom trenutku, sa onim sto smo ucili do sada).

Problem /home/rich/ellcc-host/ellcc/build/src/musl/src/env/__libc_start_main.c: No such file or directory:

Kao sto je vec receno iz fje main se vracamo u fju __libc_start_main.c, ukoliko instrukciju jr $ra izvrsimo sa “si: u gdb-u dobit cemo sljedeci ispis:
(gdb)
0x7675e22c in __libc_start_main (main=0x410564, argc=4261216, argv=0x1) at /home/rich/ellcc-host/ellcc/build/src/musl/src/env/__libc_start_main.c:74
74 /home/rich/ellcc-host/ellcc/build/src/musl/src/env/__libc_start_main.c: No such file or directory.
(gdb)
0x7675e230 74 in /home/rich/ellcc-host/ellcc/build/src/musl/src/env/__libc_start_main.c

Ovo ustvari nije nikakva greska, jer mi i nemamo izvorni fajl u kome se nalazi fja __libc_start_main. Bez simbolickih informacija iz fajla ne mozemo debuggirati ovu fju, i zato dobijamo ovaj ispis.
Ukoliko instrukciju jr $ra izvrsimo sa “ni” u gdb-u necemo ulaziti u fju __libc_start_main. Fja ce se u cjelosti izvrsiti i dobit cemo sljedeci ispis:

(gdb)
[Inferior 1 (Remote target) exited normally]