

Domaći zadatak IP-1 [15 bodova] (Utorak 07H)

Kreirati softverski karakter drajver koji modeluje Aritmetičku Logičku jedinicu 'alu' za rad sa pozitivnim celobrojnim vrednostima. Dodela upravljačkih brojeva drajveru je dinamička. Skladno tome, napisati odgovarajuće *init* i *exit* funkcije, koje uključuju/isključuju modul iz kernela pri pozivu komandi *insmod* i *rmmmod*. Preklopiti funkcije za rad sa fajlovima: *open*, *release*, *read* i *write* na sledeći način:

1) [3b] Slanje komandi:

Postoje četiri osmobarbitna registra za operande: 'regA', 'regB', 'regC' i 'regD', jedan osmobarbitni registar za čuvanje rezultata 'result' i jedan statusni registar 'carry' - indikator prekoračenja opsega. Postavljanje nekog od registara za operande na određenu vrednost se vrši u dekadnom formatu na sledeći način:

Format komande:

```
echo reg X =YYY > /dev/alu gde su: X ={a,b,c,d} YYY – vrednost od 0-255
```

Na primer:

```
echo regA=64 > /dev/alu
```

će u registar 'regA' upisati vrednost 64.

```
echo regB=25 > /dev/alu
```

će u registar 'regB' upisati vrednost 25.

Slanje operacije koja treba da se izvrši se radi na sledeći način:

Format komande:

```
echo reg X ? reg Y gde su: X , Y ={A,B,C,D} ? ={+,-,*,/}
```

Na primer:

```
echo regA + regB > /dev/alu
```

će u registar 'result' upisati zbir stanja registara regA i regB, za prethodni primer

će to biti 64+25=89

2) [3b] Čitanje rezultata neke operacije se vrši komandom:

```
cat /dev/alu
```

pri čemu se ispisuje stanje registra 'result' u dekadnom formatu a zatim i stanje statusnog registra 'carry'. Nakon poziva ove komande se u terminalu ispisuje rezultat prethodno izvršene aritmetičke operacije.

Za primer od gore:

```
cat /dev/alu
```

```
89 0
```

Što indicira da je rezultat operacije 89 i da se nije desilo prekoračenje opsega carry = 0;

3) [1b] Konfiguracija Implementirati pomoćnu naredbu:

```
echo format= aaa > dev/alu gde je: aaa ={dec,hex,bin}
```

Podesiti da se pri čitanju ispisuje format podešen pomoću prethodne naredbe.

4.1) [1b] Blokiranje : Pokušaj uzastopnog slanja više komandi će rezultovati blokiranjem procesa sve dok se ne pročita prethodno proračunato rešenje iz registra 'result' pomoću *cat* komande.

4.2) [2b] Zaštita : Pomoću semafora zaštititi deljene resurse kako izvršavanje u preemptive i SMP sistemima ne bi rezultovalo u potencijalnim hazardima.

5.1) [3b] Aplikacija : Napraviti jednostavnu C aplikaciju za rad sa drajverom. Aplikacija traži od korisnika da unese matematički izraz sa maksimalno 4 celobrojna operanda u opsegu [0-255], a zatim rešava taj izraz isključivo pozivima drajvera /dev/alu te rezultat ispisuje u terminalu.

Unesite izraz:

$225-100*2+25$

Rešenje je:

50

Dok god korisnik unosi matematičke izraze, aplikacija ponovo traži od korisnika da unese izraz. Tek kada korisnik umesto izraza unese reč "exit", aplikacija se terminira.

5.2) [2b] Proširenje aplikacije Proširiti aplikaciju na takav način da je moguće uneti proizvoljno dug matematički izraz uz podršku korišćenja zagrada (ne treba implementirati podršku za ugnježdene zagrade).

Na primer:

$225-(100+2)+25-(3*2)$

Napomena:

Rešenje poslati u vidu git linka.

Obezbediti Makefile za drajver i aplikaciju.

Zadatak je moguće raditi u paru.