

Zadatak:

Realizovati assemblerske potprograme koji čine drajver bitmapiranog ekrana terminala računara treće generacije. Sadržaj bitmapiranog ekrana se nalazi u kontinualnom bloku memorije određene promenljivom **screen**. Njegova širina i visina se nalaze u promenljivim **scrwidth** i **scrheight**, a ukupna veličina memorijskog prostora je u promenljivoj **scrszize**. Validne koordinate tačaka su $[0, \text{scrwidth}-1]$ i $[0, \text{scrheight}-1]$ za X i Y ose respektivno. Realizovani potprogrami drajvera treba da rade i sa drugim vrednostima **scrwidth** i **scrheight** promenljivih, a ne samo sa onim koje su inicijalno zadate (pri čemu treba smatrati da se njihove vrednosti neće menjati kada se program jednom pokrene).

Bitmapirani ekran podržava rad sa 4 boje: crna (0b00), crvena (0b01), zelena (0b10) i plava (0b11). Boje su kodirane sa po dva bita, pa se 4 piksela nalaze u jednom bajtu ekranske memorije (svako slovo predstavlja 1 bit):

```
AABBCCDDEEFFGGHHIIJJKKLL . . .  
MMNNOOPPQQRRSSTTUUVVXXYY . . .  
. . .
```

gde AABBCCDD predstavlja prvi bajt ekranske memorije, EEEFFGGHH je drugi bajt i tako redom. Više značajni biti odgovaraju tačkama sa manjom X koordinatom. Na primer:

AA su dva najznačajnija bita prvog bajta prve linije ekrana i odgovaraju pikselu sa koordinatama (0,0).

DD su dva najmanje značajna bita prvog bajta prve linije ekrana i odgovaraju pikselu sa koordinatama (3,0).

MM su dva najznačajnija bita prvog bajta druge linije ekrana i odgovaraju pikselu sa koordinatama (0,1).

Ako su X i Y kooridinate tačke, tada se adresa bajta u ekranskoj memoriji u kome se tačka nalazi može dobiti na sledeći način:

$$\text{adresa_ekrana} + (X/4) + (Y * \text{širina_ekrana}/4)$$

Deljenje sa 4 je celobrojno i radi se zato što se u jednom bajtu nalaze 4 piksela (i za X i za Y koordinatu; Y koordinata se množi i sa brojem tačaka po X osi, kako bi se dobila odgovarajuća linija ekrana).

Pozicija (prvog bita) piksela unutar bajta se može dobiti od X koordinate po modulu 4: $X\%4$, što se lako (i brže) može realizovati sa logičkim naredbama. Međutim, pošto višim bitima odgovaraju niže koordinate, pozicija bita je u stvari obrnuta, a pošto su dva bita po pikselu, onda je još i pomnožena sa 2: $(3 - X\%4) * 2$. Npr. za prvi bajt ekrana (Y=0) AABBCCDD i za X=1, formula daje vrednost 4, što znači da treba menjati 4 i 5 bit tog bajta, BB. Iscrtavanje tačke na zadatoj poziciji se može lakše odraditi prvo upisivanjem crne boje (0b00), pa onda upisivanjem boje koja se traži.

Potrebno je realizovati sledeće assemblerske potprograme koji čine drajver ekrana i omogućavaju rad sa njim:

```
void plot(int x, int y, int col);
```

Iscrtavanje tačke na zadatim koordinatama u zadatoj boji. Ukoliko je bilo X, bilo Y kooridinata pogrešna, tačku ne treba crtati. Smatrati da boja neće izlaziti iz opsega [0,3].

```
void drawrect(int x1, int y1, int x2, int y2, int col);
```

Iscrtavanje pravougaonika određenog koordinatama (x1,y1) i (x2,y2) u zadatoj boji. Smatrati da važi $x1 \leq x2$ i $y1 \leq y2$. Smatrati da boja neće izlaziti iz opsega [0,3].

```
void fillrect(int x1, int y1, int x2, int y2, int col);
```

Iscrtavanje popunjenog pravougaonika određenog koordinatama (x1,y1) i (x2,y2) u zadatoj boji. Smatrati da važi

$x1 \leq x2$ i $y1 \leq y2$. Smatrati da boja neće izlaziti iz opsega [0,3].

void sprite(int x, int y, unsigned char *data);

Iscrtavanje sličice (sprajta) na zadatim koordinatama. Sprajt je sličica dimenzija 16x16 piksela zadata nizom od 64 bajta, sa organizacijom boja istom kao i ekranska memorija:

AABBCCDDEEFFGGHHIIJJKKLLMMNNOOPP

QRRSSTTUUVVWVXX . . .

. . .

Prva četiri bajta čine prvi red od 16 piksela, druga četiri bajta drugi red i tako dalje. Prilikom iscrtavanja sprajta, crnu boju ne treba iscrtavati (crna se za sprajtove smatra transparentnom bojom).

Potprogrami **drawrect**, **fillrect** i **sprite** se mogu lakše realizovati ako se naprave tako da koriste prethodno realizovani potprogram **plot**.

Za testiranje i simulaciju izlaza ekrana je dat glavni.c. Program sadrži nekoliko funkcija koje omogućavaju simulaciju bitmapiranog ekrana, kao i nekoliko funkcija koje realizuju test primere. **Preporučuje se da se asemblerski potprogrami realizuju postepeno, shodno redosledu testova.** Kako izlaz testova treba da izgleda se može videti u fajlovima test1 do test7 (testovi 6 i 7 su jednostavne animacije koje ilustruju upotrebu potprograma drajvera terminala). **Za kompajliranje programa treba koristiti gcc . sh skript**, pošto treba uneti veći broj parametara pri pozivu gcc kompajlera.

Kod koji se nalazi unutar plot potprograma inicijalno postavlja tačku u prvi i poslednji bajt ekrana i tu je samo za ilustraciju kako se ekranu može pristupiti. Treba ga obrisati i napisati kod po gornjoj specifikaciji.