

Operativni Sistemi

Video memorija; veza sa C-om

Video memorija

- Počinje na adresi B800h
- Redom svaki karakter opisan sa 2 bajta
 - ASCII vrednost slova koje treba ispisati
 - Kod boje
- Boja se kodira iz tri dela:
 - Prva (najniža) 4 bita predstavljaju boju slova
 - Naredna (središnja) 3 bita predstavljaju boju pozadine
 - Poslednji (najviši) bit je flag za treperenje

Adresiranje

- Određivaćemo bajt na koji treba upisivati vrednost pomoću para [ES:BX] (ovo je naš „pokazivač“)
- ES je „bitniji“ deo adrese. Tu će biti upisana bazna adresa, a BX će biti offset.
- Dakle, u programu se ES postavlja na B800h, a BX se inkrementira. Jedan inkrement BX pomera naš „pokazivač“ unapred za jedan bajt u memoriji.

Prošetaj smajlija

- Napisati program u assembleru koji šeta smajlija po ekranu.
 - w – gore
 - a – levo
 - s – dole
 - d – desno
 - Esc – izlazak iz programa
- ASCII kodovi:
 - Smajli = 1
 - w = 119
 - a = 97
 - s = 115
 - d = 100
 - Esc = 33

Varijacije

- Varijacija 1: Napraviti da se ništa ne dešava pritiskom na neko dugme koje nije jedno od ovih 5
- Varijacija 2: Paziti na „wrap-ovanje“, tj. da kad smajli ode do kraja levo, desno, gore ili dole, da izađe na dobrom mestu sa suprotne strane ekrana
- Varijacija 3: Rainbow mode: smajli menja boje dok se kreće

Veza sa c-om

- Može da bude korisno da povežemo naš assembler kod sa nekim algoritamski komplikovanijim kodom radi testiranja
- Procedura se svodi na to da se .c i .asm fajl odvojeno kompajluju kao objektni fajlovi, i onda linkuju u konačni .exe. Koristićemo gcc kompajler za kompajlovanje .c fajlova i linkovanje
- gcc kompajler se dobija uz MinGW. MinGW može da se instalira samostalno, a distribuira se i uz Dev-Cpp i uz Code::Blox.
- Path varijabla mora da sadrži putanju do bin foldera MinGW-a

Putanje

- Sistemska PATH promenljiva mora da sadrži putanje do gcc i nasm
- Sistemska LD_LIBRARY_PATH promenljiva mora da sadrži putanju do lib foldera MinGW
- Podešavanje promenljivih:
 - Desni klik na My Computer
 - Properties
 - Kartica Advanced
 - Dugme Environment Variables
 - Dugme „New“ za novu promenljivu, ili „Edit“ za menjanje već postojeće

Primer linkovanja c i assembler koda

- Kompajlovanje .c fajla
 - `gcc -o test_printf.o -c test_printf.c`
- Kompajlovanje .asm fajla
 - `nasm -f win32 -o call_printf.o call_printf.asm`
- Linkovanje
 - `gcc test_printf.o call_printf.o -o call_printf.exe`
- Napomena: ovaj primer izvršavati pod Windows okruženjem (unutar command prompt), ne DOS! Gcc nije portovan na DOS. Neophodno je koristiti nasm kompajler za Windows.

Sitni (ne)bitni detalji

- Pod windowsom, sve C funkcije kada se kompajliraju dobijaju dodatnu `_` na početku imena. Dakle `printf` se poziva kao `_printf`. Ako želimo da C vidi našu f-ju, njeno ime mora početi sa `_` i iz C koda f-ju pozivamo bez `_`.
- Argumenti funkcije koju pozivamo se stavljaju na stek obrnutim redosledom od onog koji bismo pisali u C-u.
- Kada C funkcija koju smo pozvali iz assembler-a završi sa radom, moramo da skinemo argumente sa steka.

Zadatak za vežbu kontrole toka

- Napisati program koji ispisuje na ekranu unapred zadat dvocifren integer broj. Funkcija treba da radi tako što se broj koji treba ispisati prvo upiše u `ax` (nema potrebe da se učitava sa tastature).
- Varijacija 1: Napisati program koji detektuje da li je broj jednocifren, dvocifren ili trocifren, pa ga onda tretira različito zavisno od toga (podrazumeva se da unos neće imati više od tri cifre).
- Varijacija 2: Napisati program koji na ekranu ispisuje proizvoljan broj. Pretpostaviti da nema ograničenja na dužinu broja, iako je maksimalna dužina samo 5.