

# Procedūrinio programavimo pagrindai

## 11 tema

Tikėtis, kad visas kodas bus viename faile, yra ne tik nerealistiška, bet ir nekorektiška – kodo skaidymas į modulius bei funkcijas yra būtina sąlyga, kad išskirstyti atsakomybes ir suvaldyti kodą. Šių užduočių tikslas – padėti darbo su daugiafailėmis programomis pagrindus.

Funkcija *main* yra skirta testavimui ir demonstracijai, kaip veikia jūsų parašytos funkcijos. Nei vienoje iš žemiau užduotyse aprašomų funkcijų negali būti nei skaitymo iš, nei spausdinimo į ekraną (nebent užduotyje nurodoma kitaip, pvz. *printArray* ar *main*). Specialios vartotojo sąsajos daryti nereikia – darbo rezultatas yra funkcijos ir moduliai, o ne *main* esantis kodas.

Naudojamo kompiliatoriaus parinktys (angl. compiler options), leidžiančias vykdyti žemiau nurodytus veiksmus, kurti bibliotekas ir valdyti transliavimo procesą, rasite kompiliatoriaus dokumentacijoje. Jei naudojate GCC, galite pasinaudoti paskaitos metu pateikta informacija. Jei jūsų kūrimo aplinka prašomų failų sukurti neleidžia, praneškite apie tai pratybų dėstytojiui.

Atliktas užduotis įkelkite į VU VMA, laikydamiesi pateiktų nurodymų.

### Užduotis 1.

Sukurkite modulį, atskirame faile *file.c* apibrėždami dvi funkcijas: *saveToFile*, leidžiančią sveikų skaičių masyvo turinį išsaugoti binariname faile, ir *loadFromFile*, leidžiančią iš to failo jį užkrauti. Kad užkraunant žinoti kiek elementų buvo saugoma masyve, išsaugodami duomenis binarinio failo pradžioje išsaugokite to masyvo dydį. Šių funkcijų prototipus pateikite antraštės faile *file.h*, kurį būtinai apsaugokite nuo pakartotinio įtraukimo (angl. include guard). Kitame faile *main.c* įtraukite *file.h* ir apibrėžkite dvi naujas funkcijas: *fillArray*, leidžiančią užpildyti masyvą atsitiktiniais duomenimis, ir *printArray*, leidžiančią atspausdinti visus to masyvo elementus į ekraną. Faile *main.c* sukurkite funkciją *main* ir patikrinkite savo parašytas funkcijas tokiu būdu: 1) sugeneruokite masyvą *m1* 2) atspausdinkite *m1* 3) išsaugokite *m1* faile 4) užkraukite iš failo masyvą *m2* 5) atspausdinkite *m2*. Jei abiejų masyvų turinys sutampa (kas reiškia, jog funkcijos veikia korektiškai), parašykite programą, kuri spausdina tekstą „all good“ į ekraną, jei masyvų turinys sutampa, ir „something’s wrong“, jei masyvų turinys skiriasi.

### Užduotis 2.

Papildykite *file.h* ir *file.c* esantį modulį dviem skaitliukais (globalūs kintamieji), leidžiančiais įsiminti, kiek kartų masyvas buvo saugomas į failą ir kiek kartų buvo iš failo nuskaitoma. Patikrinkite, jog tie skaitliukai skaičiuoja operacijas korektiškai, tokiu būdu: 1) sugeneruokite masyvus *m1*, *m2* ir *m3* su atsitiktiniais duomenimis 2) išsaugokite masyvą *m1* į failą 3) į tą patį failą („ant viršaus“) išsaugokite masyvą *m3* 4) užkraukite iš to failo masyvą *m1* 5) išsaugokite kitame faile masyvą *m2* 6) užkraukite iš to (kito) failo masyvą *m3* 7) užkraukite iš pirmo (ankstesnio) failo masyvą *m2*. Parašykite programą, kuri spausdina „all good“, jei įvykdžius minėtą scenarijų skaitliukas *saveCount* lygus 3 ir skaitliukas *loadCount* lygus 3, o kitu atveju – spausdinkite „something’s wrong“.

### Užduotis 3.

Kompiliuodami per komandinę eilutę, sugeneruokite šiuos failus: 1) gaukite *file.i*, apdoroję *file.c* preprocesoriumi 2) gaukite *file.s* su assemblerio kodu, atitinkančiu *file.c* ir/ar *file.i* esantį kodą 3) gaukite *file.o*, kuriame būtų sukompiliuotas *file.c* kodas 4-6) analogiškai gaukite failus *main.i*, *main.s* ir *main.o*, iš failo *main.c* ir 7) gaukite programos vykdomąjį failą (*main.exe* arba jo analogą), surišdami (angl. linking) failus *file.o* ir *main.o*. Pasižiūrėkite, kaip atrodo kiekvienas iš gautų failų, per tekstinį redaktorių. Kurie failai yra tekstiniai, skaitomi/suprantami žmogui, o kurie ne (skirti „kompiuteriui“)?

#### **Užduotis 4.**

Sukurkite statinę biblioteką *libfile.a* (arba *libfile.lib*), į kurios vidų įdėkite failą *file.o*. Pagaminkite vykdomąjį failą (*main-static.exe* arba jo analogą), naudodamiesi ta sukurta statine biblioteka. Įsitikinkite, kad programa *main-static* veikia korektiškai.

#### **Užduotis 5.**

Sukurkite dinaminę biblioteką *libfile.dll* (arba *libfile.so*), į kurios vidų įdėkite failą *file.o*. Pagaminkite vykdomąjį failą (*main-dynamic.exe* arba jo analogą), kuris galėtų dirbti su ta dinamine biblioteka. Įsitikinkite, kad programa *main-dynamic* veikia korektiškai.

#### **Papildomos užduotys.**

#### **Užduotis 6.**

Modifikuokite skaitliukus (užduotis 2) taip, kad jie skaičiuotų ne tik saugojimo ir užkrovimo operacijas bendrai, bet ir kiekvienam failui atskirai. Operacijų skaičiams gauti apibrėžkite funkcijas, leidžiančias nurodyti aktualų failą. Jei failas nenurodomas (pateikiama rodyklė lygi NULL), funkcijos grąžina bendrą kiekį (sumuoja skaitliukus visiems failams).

#### **Užduotis 7.**

Atskirkite skaitliukus į atskirą modulį *counters*, su kuriuo susiekite ir modulį *file*, ir *main*. Iš naujo įvykdykite 1) užduotį 3, failams *counters.i*, *counters.s*, *counters.o* gauti 2) užduotis 4 ir 5, statinei ir dinaminei bibliotekoms gauti, kurių viduje būtų du failai, *file.o* ir *counters.o*. Atnaujinkite ir ištestuokite programą *main* su naujomis bibliotekomis.

#### **Užduotis 8.**

Modifikuokite užduotyje 1 nurodytas funkcijas, kad faile būtų galima išsaugoti ir užkrauti ne sveikųjų skaičių, o bet kokių (fiksoto dydžio) elementų masyvą. Operacijų korektiškumui užtikrinti, binarinio failo pradžioje saugokite ne tik masyvo, bet ir jo elemento dydį, ir jei užkrovimo metu nurodomas dydis neatitinka išsaugotojo – nuskaitymo nevykdykite.