1. Atidaryti faila…, nuskaityti TIK paskutini failo baito…(neskaitant viso likusio failo) ir atspausdinti ji

pabandytø pirmu komandinës eilutës argumentu nurodytà failà "primapinti" á proceso adresø erdvæ;

jei pavyko - iðkviestø ðio failo N-ojo baito reikðmæ (ið primapinto regiono) á ekranà (kaip sveikà skaièiø), N nurodomas antru komandinës eilutës argumentu (N prasmë kaip C masyvo indekso, t.y. N=0 -> pirmas baitas);

programa turi apdoroti galimas klaidas.

PASTABA: nebûtina mapinti viso failo (t.y. ar primapinsit visà, ar tik dalá failo - átakos ávertinimui neturës).

1. priimtø 3 komandinës eilutës argumentus

pakeistø darbiná katalogà á pirmu argumentu norodytà

atidarytø antru komandinës eilutës argumentu nurodytà failà tik skaitymui su fopen() funkcija;

atidarytø treèiu argumentu nurodytà failà raðymui

nuskaitytø ið pirmo failo su fread() ir áraðytø á antrà failà su fwrite()1024 baitus (jei tiek yra pirmame faile)

atspausdintø pirmo ir paskutinio nukopijuotø baitø reikðmes (intervalas 0..255, arba -128..127)

programa turi apdoroti galimas klaidas.

**LAB 3-2**

**getcwd(), chdir(), fchdir()**

1. gautų ir atspausdintų einamo katalogo vardą su getcwd() (ir kviečiant getcwd() naudotųpathconf(".",\_PC\_PATH\_MAX) grąžinamą reikšmę)
2. atidarytų einamą katalogą su open(), įsimintų ir atspausdintų jo deskriptorių
3. nueitų į /tmp katalogą su chdir()
4. patikrintų su getcwd() ir atspausdintų koks dabar yra einamasis katalogas
5. grįžtų į 2-ame žingsnyje atidarytą katalogą su fchdir() (ir patikrintų/parodytų, kad tikrai grįžo)

**opendir(), fdopendir(), readdir(), closedir()**

1. atidarytų einamą katalogą su opendir() arba fdopendir();
2. cikle nuskaitytų visus katalogo įrašus su readdir() ir išvestų kiekvieno įrašo *i-node* numerį (direntstruktūros d\_ino laukas) ir failo vardą (dirent struktūros d\_name laukas);
3. uždarytų katalogą su closedir();

**stat(), fstat()**

* kuri su stat() ar fstat() gautų informaciją apie komandinės elutės parametru jai nurodytą failą, katalogą ar kt. ir išvestų į ekraną statstruktūros turinį:
  + savininko ID
  + dydį
  + i-node numerį
  + leidimus
  + failo tipą (katalogas/failas/konvejeris/soketas...)

**statvfs(), fstatvfs()**

Nukopijuokite *loginas\_stat01.c* į **loginas\_statvfs01.c** ir papildykite, kad papildomai būtų išvedama irstatvfs() arba fstatvfs() grąžinamos struktūros statvfs informacija:

* failų sistemos bloko dydis
* failų sistemos tipas (f\_basetype, nestandartinis laukas statvfs struktūroje, žr. man statvfs ant*os.if.ktu.lt*)
* failų sistemos dydis
* maksimalų failo kelio/vardo ilgis

**nftw()**

* Sukurkite ir išbandykite programą **loginas\_nftw02.c**, kuri su nftw() išvaikščiotų Jūsų namų katalogą ir atspausdintų visų jame esančių failų pavadinimus (t. y. pradėtų paiešką nuo Jūsų namų katalogo ir naudotų FTW\_PHYS, kad neišeitų iš jo radus simbolinę nuorodą).
* Įkelkite programos tekstą į Moodle.

**LAB 3-3**

**Sinchroninis I/O**

Sukurkite programą **loginas\_rw01.c**, kuri:

1. atidarytų komandinėje eilutėje nurodytą failą tik skaitymui su open();
2. atidarytų kitą komandinėje eilutėje nurodytą failą tik rašymui (sukurtų, jei nėra, išvalytų turinį jei jau yra);
3. nukopijuotų iš skaitomo failo į rašomą komandinėje eilutėje nurodytą baitų skiačių (jei tiek baitų nėra – tiek kiek yra, t. y. visą failą) naudojant read() ir write();
4. uždarytų abu failus su close().

Sukurkite programą **loginas\_seek01.c**, kuri:

1. sukurtų failą (su open() ar creat());
2. nueitų į 1MB gilyn į failą su lseek();
3. įrašytų 1 baitą;
4. uždarytų failą su close().

**Buferizuotas I/O**

Nukopijuokite loginas\_rw01.c į **loginas\_frw01.c** ir pakeiskite, kad vietoj open() būtų naudojama fopen(), vietojclose() – fclose(), vietoj read() – fread(), vietoj write() – fwrite().  
Išbandykite naują programą. Turėtų gautis toks pat rezultatas.  
Kuo skiriasi fgetc() ir getc(): ?

**Asinchroninis I/O**

* Sukurkite programą **loginas\_aio02.c** (galite naudoti pavyzdžio ar savo anksčiau sukurtų programų fragmentus), kuri iš /dev/urandom su aio\_read() nuskaitytų 1MB duomenų (t. y. tiek kiek prašoma).
  1. bus reikalingi pakartotiniai aio\_read() iškvietimai;
  2. reikės keisti adresą kur rašyti duomenis, kad neperrašytų ant jau nuskaitytų.

**Failų „mapinimas“ į RAM**

* Sukurkite programa **loginas\_mmap02.c**, kuri nukopijuotų failus naudojant mmap() (kad būtų paprasčiau laikykime, kad failų dydžiai iki 100MB, t. y. abu telpa į 32bit proceso erdvę):
  1. atidarytų ir prijungtų 2 programos argumentais nurodytus failus su mmap() (vieną iš jų tik skaitymui, tik skaitomo failo dydį galite sužinoti su fstat() funkcija)
  2. nukopijuotų vieno failo turinį į kitą (su memcpy() ar paprastu ciklu)
  3. atjungtų abu failus
  4. uždarytų abu deskriptorius

**LAB 3-4**

Informacija apie procesams taikomus apribojimus

* Sukurkite programą **loginas\_cpulimit01.c**, kuri nustatytu CPU limitą 1s (RLIMIT\_CPU) ir patikrinkite, ar limitas suveikia (nustatę limitą užsukite amžiną ciką su skaitliuku).  
  Suveikus limitui programa turėtų mest *core*.
* Kiek iteracijų padarė amžinas ciklas: ? (galite tai sužinoti iš core failo; jei įtariate, kad skaitliukas persipildė – pabandykite perkompiliuoti į 64bit su kompiliatoriaus parametru -m64 arba sumažinti CPU laiko limitą).
* Pataisykite, kad programa nemestų *core* (nustatykite RLIMIT\_CORE limitą į 0)

Programos darbo užbaigimas

* Sukurkite programą **loginas\_exit01.c** ir joje su atexit() priregistruokite kelias savo funkcijas (bent 3). Pagal programai paduotą parametrą išeikite su \_Exit(), exit(), abort() arba return.
* Kaip veikia: iškviečia priregistruotas funkcijas, neiškviečia, iškviečia ne visas, kokia tvarka iškviečia?
* Kaip keičiasi priregistruotų funkcijų iškvietimas, jei iš programos išeinama su \_Exit(), exit(), return, abort(): ? (kuriais atvejais iškviečiama, kuriais ne)

Dinaminis užkrovimas

* Perdarykite bet kurias dvi anksčiau darytas C programas į **loginas\_testlib02a.c**, **loginas\_testlib02b.c** ir**loginas\_testlib02.h**, taip, kad galėtumėte sukurti dinamines bibliotekas libloginas02a.so irlibloginas02b.so ir kad kiekviena iš jų turėtų bent vieną funkciją tokiu pat vardu, parametrais ir grąžinama reikšme (pvz.: int vp\_testlib(int a); ). Kitaip sakant abiejų bibliotekų kompiliavimui turėtų tikti tas patsloginas\_testlib02.h header failas.