

Sentenced sana- ja virkelaskuri

C-ohjelmoinnin harjoitustyö

26.10.2011

Markus Koski

Hansi Keijonen

Sentenced sana- ja virkelaskuri

1. Ohjelman kuvaus

Sentenced laskee tekstistä sanat, virkkeet ja laskee sanoja/virke -jakauman. Teksti voidaan syöttää ohjelmalle joko näppäimistöltä tai antamalla syötetiedosto. Tulokset tulostuvat stdout:iin ja tiedostoon, tulostiedoston nimen voi myös halutessaan valita itse.

2. Ohjelman käyttö/käynnistäminen

Jos ohjelmaa ei ole käännetty, kirjoita komentorivillä

`make`

Ohjelma käynnistetään komennolla

`./sentenced.exe -optiot`

Käynnistyskomento-optiot voivat olla:

<i>ei optiota</i>	tulostaa lyhyen ohjeen optioiden käytöstä
<code>-h</code>	tulostaa readme-tiedoston
<code>-s</code>	teksti syötetään näppäimistöltä, <ctrl>-d lopettaa
<code>-i<tiedostonimi></code>	teksti luetaan tiedostosta
<code>-o<tiedostonimi></code>	vaihdetaan tulostiedoston nimi, oletuksena results.txt

Esimerkki: ./sentenced.exe -iteksti.txt -otulos.txt

3. Ohjelman rakenne ja toiminta

3.1. Tiedostokuvaukset

Seuraavassa ainoastaan tiedostojen yleiskuvaukset. Tiedostojen sisältämät toiminnot ja funktiot selviävät yksityiskohtaisesti lähdetiedostoista.

3.1.1. main.c

Tiedosto main.c lukee komentoriviparametrit ja ohjaa ohjelman toiminnan niiden mukaisesti (kts 2.)

Tiedostossa on myös syöte- ja tulostustiedostojen avaaminen.

3.1.2. misc.c

Tiedostossa misc.c on main.c:ssä kutsuttavia apuoperaatioita, kuten ohjeen tulostaminen, virheen käsittelyä sekä readme-tiedoston avausoperaatio.

3.1.3. avltree.c

Tiedostossa avltree.c määritellään tietorakenne AVL-puu. AVL-puun perusoperaatioista on toteutettu solmun lisäys, puun tuhoaminen sekä solmujen tulostus. AVL-puun referenssinä käytettiin Tietorakenne -kurssin luentomonistetta keväältä 2011.

3.1.4. read.c

Tiedosto read.c lukee tekstiä (tiedostosta tai näppäimistösyötteestä) sana kerrallaan, poistaa ylimääräiset merkit sanojen aluista ja loppuista ja lisää ne AVL-puuhun. Tiedostossa myös lasketaan sanat, virkkeet ja sanat/virke -jakauma ja tulostetaan tulokset tiedostoon ja stdout:iin.

3.1.5. tests.c

Tiedosto tests.c sisältää testifunktiot avltree.c:n testaamiseen. Se muodostaa erilaisia syötteiden erikoistapauksia, lisää ne puuhun ja validoi puun sitten AVL-puuehtojen mukaan.

3.1.6. sentenced.h

Otsikkotiedostossa sentenced.h on määritelty struktuuri ja useita funktioita, joita voidaan käyttää kaikkialla kooditiedostoissa.

3.2. Ohjelman toiminta pääpiirteissään

Kun ohjelma käynnistetään ilman optioita, kutsuu main.c misc.c:ssä määriteltyä funktiota info(), joka tulostaa ruudulle ohjeen optioiden käytöstä.

Kun ohjelma käynnistetään optiolla -s, kutsuu main.c read.c:n read_words() -funktiota, joka lukee tekstin näppäimistöltä, lisää sanat puuhun avltree.c:n add_word() -funktiolla ja tulostaa tulokset stdout:iin.

Kun ohjelma käynnistetään optiolla -i<tiedostonimi>, kutsuu main.c read.c:n read_words() -funktiota, joka lukee tekstin nimetystä tiedostosta, lisää sanat puuhun avltree.c:n add_word() -funktiolla ja tulostaa tulokset stdout:iin.

Kun ohjelma käynnistetään optiolla -o<tiedostonimi>, asetetaan annettu tiedostonimi tulostiedoston nimeksi.

Kun ohjelma käynnistetään optiolla -h, kutsuu main.c misc.c:n help() funktiota, joka tulostaa readme.txt:n.

Tiedosto avltree.c:ssä on kaikki puun käsittelyyn tarvittavat funktiot. add_word() lisää sanan ja kutsuu tarvittavia tasapainotusoperaatioita. Ohjelman suorituksen lopussa tulee kutsua destroy_tree-funktiota, joka tuhoaa puun ja vapauttaa sen varaaman muistin.

Tests.c luo erilaisia puita, jotka sitten validoidaan validate_tree() funktion avulla. Testien ajaminen selitetty seuraavassa kappaleessa.

4. Ohjelman testaus

Ohjelmasta on testattu tietorakenteen toiminta ja muistivuodot.

Tietorakenne voidaan testata poistamalla kommentti main.c:ssä funktiokutsun run_test() edestä ja ajamalla ohjelma ilman optioita. AVL-puurakenteesta tarkistetaan jokaisen solmun osalta, täyttääkö se AVL -ehdot: alipuiden korkeusero -1, 0 tai 1 sekä sen, onko vasempana lapsisolmuna aina aakkosissa pienempi merkkijono ja oikeana lapsena vastaavasti aakkosissa suurempi merkkijono.

Testifunktio syötää ensiksi puuhun sanoja kasvavassa järjestyksessä, validoi puun ja ilmoittaa tuloksen. Seuraavaksi sanoja syötetään alenevassa järjestykseen, validoidaan ja tulostetaan tulokset. Kolmas testitapaus on ainoastaan yhden sanan tapaus.

Testauksen yksityiskohdat selviävät parhaiten kooditiedostossa.

Ohjelma testattiin muistivuotojen varalta ajamalla se Valgrind – ohjelmalla.