

Tuukka Kurtti
tuukkak123@gmail.com
Oulun yliopisto
28.12.2014

1. Yleistä

Tässä tiedostossa esittelen hieman tarkemmin tätä projektia varten tekemäni syötteen-lukufunktiota ja sen käyttöä. Alussa kaikki syötteen lukuun liittyvät funktiot olivat samassa paikassa osana pankkiautomaatin lähdekoodia, mutta huomasin, että se on kokonaisuus, jota voidaan käyttää yleisemmin useissa eri projekteissa. Niinpä erittelin kaikki syöte-funktioon liittyvän koodin erilliseen syote.c -tiedostoon. Jos tätä laajennusta aikoo jatkaa, olisi ehkä hyvä tehdä erillinen header tiedosto, jotta koodia ei tarvitse sisällyttää includella kokonaan osaksi omaa lähdekoodia.

2. Yleinen käyttö

Funktiokutsu on toteutettu makron välityksellä, ja näyttää seuraavanlaiselta:

syote(tietotyyppi, viesti, lisäehdot)

- Tietotyyppi kertoo, mitä tietotyyppiä olevaa arvoa halutaan käyttäjältä lukea. Tuetut tietotyypit ovat int, double ja string.
- Viesti on merkkijono (char[]), joka tulostetaan käyttäjälle syötekentän eteen.
- Lisäehdoilla voidaan määrittää tarkemmin, minkälaisen syötteen haluaa käyttäjän antavan. Lisäehtoja voi olla yksi tai useampi ja niiden toimintaa on selitetty tarkemmin kappaleessa 3.

Funktion palautusarvon tyyppi riippuu ensimmäisestä parametrasta. Tietotyypin ollessa int tai double on palautusarvo sitä vastaava. Tietotyypin ollessa string palautetaan osoitin dynaamisesti varatun merkkijonon alkuun. Kun tätä merkkijono ei enää käytetä, tulee se vapauttaa free() -funktiolla. Tietotyyppi string tarkoittaa dynaamisesti varattua merkkijonoa, jonka tietotyyppi on käytännössä sama kuin char*, mutta makron nimessä ei voi olla asteriskimerkkiä.

Lyhyt esimerkki funktion käytöstä:

```
int arvo;
string jono;
arvo = syote( int, "Anna positiivinen luku > ", POSITIIVINEN );
jono = syote( string, "Syota merkkijono > ", MITA_TAHANSA );
free( jono );
```

3. Lisäehdot

Lisäehdoilla voidaan määrittää käyttäjän antamalle syötteelle ehtoja, jotka annetun syöteen tulee täyttää. Jos ehto ei täyty, tulostetaan siitä käyttäjälle virheilmoitus ja samaa syötettä pyydetään uudestaan, kunnes ehto täyttyy.

3.1. Luettelo lisäehdoista

Kaikki tietotyypit:

- MITA_TAHANSA: Ei erityistä lisäehtoa, mikä tahansa kysyttyä tietotyyppiä oleva syöte kelpaa.
- SALLITUT_ARVOT(lista): Ainoastaan listassa olevat syötteet kelpaavat.
- MONIEHTO(lista): Lista useasta lisäehdosta, jotka kaikki täytyy toteutua. Ehtojen toteutuminen tarkistetaan järjestyksessä ja virheilmoitus tulostetaan ensimmäisestä toteutumatta jääneestä ehdosta.

Tietotyypit int ja double:

- POSITIIVINEN: Syöteen on oltava positiivinen luku.
- NEGATIIVINEN: Syöteen on oltava negatiivinen luku.
- SUUREMPI_KUIN(a): Syöteen on oltava suurempi kuin luku a.
- PIENempi_KUIN(a): Syöteen on oltava pienempi kuin luku a.
- VALISSA(a,b): Syöteen on oltava arvojen a ja b rajaamalla välillä.

Tietotyyppi string:

- KIELLETYT_MERKIT(jono): Syöte ei saa sisältää merkkejä, jotka ovat merkkijonossa jono.
- SALLITUT_MERKIT(jono): Syöte saa sisältää ainoastaa merkkejä, jotka ovat merkkijonossa jono.
- PIDempi_KUIN(a): Syöteen on oltava pidempi kuin luku a.
- LYHYempi_KUIN(a): Syöteen on oltava lyhyempi kuin luku a.
- PITUUS(a,b): Syöteen pituuden on oltava arvojen a ja b rajaamalla välillä.

Moniehdon käyttäminen vaatii erilaista syntaksia. Jokainen listassa oleva ehto täytyy ympäröidä aaltosulkeilla, sekä jokaisen ehdon eteen täytyy kirjoittaa kyseessä oleva tietotyyppi. Vertaa seuraavia rivejä, jotka tekevät täsmälleen saman asian:

```
arvo = syote( int, "LUKU: ", POSITIIVINEN );  
arvo = syote( int, "LUKU: ", MONIEHTO( {intPOSITIIVINEN} ) );
```

Moniehtoa kannattaa kuitenkin käyttää vain jos ehtoja on useita, kuten näin:

```
jono = syote( string, "JONO: ", MONIEHTO(  
    {stringLYHYempi_KUIN(10)}, {stringKIELLETYT_MERKIT("-.,")} ) );
```

4. Hieman syvempää tietoa laajennuksen toiminnasta

syote.c funktiot on jaettu lukufunktioiksi ja tarkastusfunktioiksi. Jokaista tietotyyppiä varten on oma lukufunktio muotoa lue_<tietotyyppi>. syote-makro huolehtii siitä, että oikeaa funktiota kutsutaan.

Kaikkien tarkastusfunktioiden parametrien ja palautusarvojen on oltava samaa tietotyyppiä, jotta niihin kaikkiin voidaan viitata samantyyppisellä funktio-osoittimella. Jotta tarkastusfunktiot olisivat mahdollisimman yleiskäyttöisiä ja riippumattomia tietotyyppistä, ovat niiden parametrit tietotyyppiä void*. Tällaista tietotyyppiä luetaan seuraavan makron avulla:

```
poke( tietotyyppi, osoitin, indeksi )  
    mikä tarkoittaa samaa kuin osoitin[indeksi], kun osoitin on tyyppiä tietotyyppi.
```

Periaatteessa syote.c toiminnallisuutta on helppo laajentaa omilla tietotyypeillä ja tarkistusfunktioilla. Esimerkiksi tietotyyppi päivämäärää varten voisi olla hyvin kätevä.