

Osa VIII

Liitteet

Liitteet

- A** Suoritusjärjestys
- B** Varatut sanat
- C** Binääri- ja heksamuoto

Osa VIII

A. Liite

Operaattoreiden suoritusjärjestys

On tärkeää ymmärtää, että operaattoreilla on prioriteettinsa, mutta ei ole olennaista muistaa tuota nimenomaista prioriteettia.

Uusi käsite Prioriteetti kuvaa suoritusjärjestystä, jonka mukaan ohjelma suorittaa operaatiot erilaisissa lausekkeissa. Jos jonkin operaattorin prioriteetti on toisen yläpuolella, se suoritetaan ensin.

Korkeamman prioriteetin operaattorit "sitovat tiukemmin" kuin matalamman tason prioriteetin operaattorit; siten ensin mainitut suoritetaan ensin. Mitä alempana jokin operaattori on seuraavassa taulukossa, sitä korkeampi sen prioriteetti on.

Taulukko A.1. Operaattoreiden suoritusjärjestys.

| Sijainti | Nimi | Operaattori |
|----------|--|---|
| 1 | Näkyvyysalue | :: |
| 2 | jäsenen valinta, indeksointi, funktio kutsut, jälkiliitteen kasvatus ja -vähennys | . -> () ++ -- |
| 3 | sizeof, etuliitteen kasvatus ja -vähennys, komplementti, ja, ei, unaarinen vähennyslasku ja yhteenlasku, osoite, viittaus, new, new[], delete, delete[], muunnos, sizeof() | ++ -- ^! + - & * |
| 4 | jäsenen valinta osoittimelle | .* ->* |
| 5 | kertolasku, jakolasku, jakojäännös | * / % |
| 6 | yhteen- ja vähennyslasku | + - |
| 7 | siirto | << >> |
| 8 | vertailu | < <= > >= |
| 9 | yhtäsuuruus, erisuuruus | == != |
| 10 | bittitason AND | & |
| 11 | bittitason XOR | |
| 12 | bittitason OR | |
| 13 | looginen AND | && |
| 14 | looginen OR | |
| 15 | ehto ilmaus | ?: |
| 16 | sijoitusoperaattorit | = *= /= %= += -= <<= >>= &= = ^= |
| 17 | throw | throw |
| 18 | pilkku | , |

Osa VIII

B. Liite

Varatut sanat

Varatut sanat on varattu kääntäjän käyttöön kielen toimesta. Et voi määritellä luokkia, muuttujia tai funktioita käyttäen yksistään näitä sanoja. Luettelomme on hieman epämääräinen, koska jotkut sanat ovat kääntäjäkohtaisia:

| | | | | |
|---------|-----------|----------|----------|----------|
| auto | break | case | catch | char |
| class | const | continue | default | delete |
| do | double | else | enum | extern |
| float | for | friend | goto | if |
| int | long | mutable | new | operator |
| private | protected | public | register | return |
| short | signed | sizeof | static | struct |
| switch | template | this | throw | typedef |
| union | unsigned | virtual | void | volatile |
| while | | | | |

Osa VIII

C. Liite

Binääri- ja heksamuoto

Opit laskemaan jo kauan sitten ja lieneekin vaikea kuvitella, miten olisit pärjännyt ilman tuota taitoa. Kun katsot lukua 145, luet sen lukuna "sataneljäkymmentäviisi" ilman kummempia miettimisiä.

Binäärin ja heksamuodon ymmärtäminen vaatii, että tutkit lukua uudelleen ja katsot sitä, ei lukuna, vaan numeron koodina.

Aloitetaan helposta: Tutki numeron kolme ja merkinnän "3" suhdetta. Numero 3 on koukero paperilla, mutta "numero kolme" on käsite. Numerokoodia 3 käytetään edustamaan numeroa.

Ero voidaan tehdä selvemmäksi esittämällä luku kolme eri tavoin: 3, |||, III ja ***. Kaikki merkitsemistavat esittävät lukuarvoa kolme.

Desimaalijärjestelmässä, jonka kantaluku on 10, käytetään numeroita 0, 1, 2, 3, 4, 5, 6, 7, 8 ja 9. Niitä on siis 10 kappaletta, kuten kantaluku (10) kertoo. Kuinka esitetään numero 10? Joku voi kuvitella, että laajennamme skaalaa ja käytämme kirjainta A esittämään lukua kymmenen tai käytämme merkintää IIIIIIII esittämään sitä. Roomalaiset käyttivät merkintää X. Arabialaisessa järjestelmässä, jota me käytämme, käytetään hyväksi numeroiden sijaintia lukujen esittämisessä. Ensimmäisessä

(oikeanpuoleisessa) sarakkeessa ovat "ykköset", seuraavassa sarakkeessa ovat "kymmenet". Siten lukua viisitoista edustaa merkintä 15, eli ykkösiä on viisi ja kymmeniä yksi.

On olemassa joitakin sääntöjä, jotka voidaan yleistää:

1. Kymmenjärjestelmä käyttää numeroita 0-9.
2. Sarakkeet ovat kymmenen potensseja: ykköset, kymmenet, sadat, jne. eli kymmenen potenssiin nolla, kymmenen potenssiin 1, kymmenen potenssiin 2 jne.
3. Jos kolmas sarake on 100, on suurin kahdella sarakkeella esitettävä luku 99. Yleistäen voidaan sanoa, että n sarakkeella voidaan esittää lukuja väliltä $0 - 10^n - 1$. Siten kolmella sarakkeella voidaan esittää luvut väliltä $0 - 10^3 - 1$ eli 0-999.

Muut kantaluvut

Ei ole sattuma, että käytämme 10-järjestelmää; onhan meillä 10 sormeä. Voisimme kuitenkin ajatella myös toisenlaista järjestelmää. Käyttäen 10-järjestelmän sääntöä (edellä) voimme kuvata 8-kantaisen järjestelmän:

1. Käytettäviä numeroita on 8 eli 0-7.
2. Sarakkeet ovat luvun 8 potensseja.
3. n sarakkeella voidaan esittää luvut väliltä $0 - 8^n - 1$.

Jotta eri lukujärjestelmän luvut voidaan erottaa, tulee kantaluku kirjoittaa alaindeksiksi luvun perään. Luku viisitoista on siten kymmenjärjestelmän lukuna 15_{10} .

Siten luku 15_{10} kantaluvun 8 järjestelmässä olisi 17_8 . Miksi 17_8 ? Numero 1 tarkoittaa 1 kertaa 8 ja 7 tarkoittaa 7 kertaa yksi. Kahdeksan plus seitsemän on sitten 15. Mietipä viittätoista asteriskia:

Ihmisellä on yleinen taipumus muodostaa kaksi ryhmää, yksi kymmenen asteriskin ryhmä ja yksi viiden asteriskin ryhmä. Se olisi desimaalilukuna 15 (1 kymmenen ja 5 ykköstä). Voisit ryhmitellä asteriskit myös seuraavasti:

**** *****

Nyt meillä on kahdeksan asteriskin ryhmä ja seitsemän asteriskin ryhmä. Se voitaisiin esittää 8-kantaisessa järjestelmässä 17_8 . Eli yksi kahdeksainen ja seitsemän ykköstä.

Kantalukuja

Voit esittää luvun 15 kymmenjärjestelmässä lukuna 15_{10} , 9-kantaisessa järjestelmässä lukuna 16_9 , 8-kantaisessa järjestelmässä lukuna 17_8 ja 7-kantaisessa järjestelmässä lukuna 21_7 . Miksi 21_7 ? 7-järjestelmässä ei ole numeroa 8. Lukuun 15 tarvitaan kaksi seitsemää ja yksi ykkönen.

Kuinka tämä prosessi yleistetään? Muuntaaksesi 10-järjestelmän luvun 7-kantaiseen järjestelmään, ajattele sarakkeita: 7-kantaisessa järjestelmässä on ykkösiä, seitsemiä, neljäkymmentäyhdeksäisiä, jne. Miksi nämä sarakkeet? Ne edustavat arvoja 7^0 , 7^1 , 7^2 , 7^3 , 7^4 jne. Luo taulukko myös itse.

| | | | |
|-------|-------|-------|-------|
| 4 | 32 | 1 | |
| 7^3 | 7^2 | 7^1 | 7^0 |
| 343 | 49 | 7 | 1 |

Ensimmäinen rivi esittää sarakenumeroa. Toinen rivi edustaa seitsemän potenssia. Kolmas rivi esittää ylemmän rivin desimaaliarvoa.

Desimaalijärjestelmän (10-järjestelmän) luku muunnetaan 7-kantaiseksi luvuksi seuraavasti: tutki lukua ja mieti, mikä olisi ensimmäinen sarake. Jos luku on esimerkiksi 200, tiedät että sarakkeen 4 arvoksi tulee nolla, mutta älä välitä siitä.

Saadaksesi selville, kuinka monta kertaa luku 49 sisältyy muunnettavaan lukuun, jaa 200 luvulla 49. Vastaus on 4, joten sijoita 4 sarakkeeseen 3 ja tutki jakojäännöstä, joka on 4. Neljään ei mahdu yhtään seitsemää, joten sarakkeeseen 2 tulee arvo 0. Ykkösiä menee lukuun 4 neljä kertaa, joten sarakkeeseen 1 tulee arvo 4. Vastaus on siis 404.

Luvun 968 muuntaminen 6-kantaiseen järjestelmään:

| | | | | |
|-------|-------|-------|-------|-------|
| 5 | 43 | 2 | 1 | |
| 6^4 | 6^3 | 6^2 | 6^1 | 6^0 |
| 1296 | 216 | 36 | 6 | 1 |

Lukuun 968 ei mene yhtään kertaa lukua 1296, joten viides sarake saa arvon 0. Jakamalla 968 luvulla 216 saadaan 4 ja jakojäännös on 104. Sarake 4 saa siis arvon 4. Jakamalla 104 luvulla 36 saadaan 2 ja jakojäännös on 32. Sarakkeen 3 arvoksi tulee siis 2. Jakamalla 32 luvulla 6 saadaan 5 ja jakojäännökseksi tulee 2. Vastaus on siten 4252_6 .

On olemassa oikotie muunnettaessa lukuja lukujärjestelmästä toiseen. Tällöin suoritetaan kertominen:

| | |
|-----------------|-----|
| $4 \cdot 216 =$ | 864 |
| $2 \cdot 36 =$ | 72 |
| $5 \cdot 6 =$ | 30 |
| $2 \cdot 1 =$ | 2 |
| Yhteensä | 968 |

Binäärijärjestelmä

Tätä ideaa on hyvä soveltaa 2-kantaisessa järjestelmässä. Käytössä on vain 2 numeroa: 0 ja 1.

| Sarake | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| Potenssi | 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| Arvo | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Muuntaaksesi luvun 88 binäärimuotoon käytä samaa proseduuria: luku 128 ei mahdu lukuun 88, joten sarake 8 on 0. Luku 64 mahtuu kerran, joten 7. sarake on 1 ja jakojäännös on 24. Luku 32 ei sisälly lukuun 24 kertaakaan, joten sarake 6 on 0. 16 menee kerran lukuun 24, joten 5. sarake on 1 ja jakojäännös on 8, johon sarakkeen 4 luku 8 menee kerran. Sarakkeeseen 4 tulee siis arvo 1. Muut sarakkeet saavat arvon 0.

Tulos:

| Sarake | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| Potenssi | 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| Arvo | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 88 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |

Testataksesi tuloksen muunna se takaisin:

$$1 \cdot 64 = 64$$

$$0 \cdot 32 = 0$$

$$1 \cdot 16 = 16$$

$$1 \cdot 8 = 8$$

$$0 \cdot 4 = 0$$

$$0 \cdot 2 = 0$$

$$0 \cdot 1 = 0$$

$$\text{Yhteensä} \quad 88$$

Miksi kantaluku 2?

Binäärijärjestelmä soveltuu hyvin esittämään tietokoneessa käsiteltävää tietoa. Tietokoneet eivät todellisuudessa tiedä mitään kirjaimista, numeroista, ohjeista tai ohjelmista. Ne ovat vain piirejä ytimeltään.

Tietokoneiden sisällä oleva toiminta ei perustu niinkään suhteellisuuteen vaan kyllä/ei-ajatteluun. Kyllä (tai tosi) voidaan esittää arvona 1 ja ei (tai epätosi) arvona 0. Yleissopimuksen mukaan 1 esittää siis arvoa kyllä, mutta yhtä hyvin voitaisiin sopia toisinkin.

Kun edellä oleva ajatus on selvänä mielessäsi, tulee binäärijärjestelmän tehokkuuskin selväksi: ykkösillä ja nollilla voidaan esittää jokaisen virtapiirin perustotuus (siellä on virtaa tai sitten ei ole). Tietokone tietää vain sen, onko jossakin kohdassa kyllä (1) vai ei (0).

Bitit, tavut ja puolikkaat

Kun tiedetään, että 1 esittää arvoa tosi ja 0 arvoa epätosi, tulevat binääriluvut hyvinkin tärkeiksi. Kun varhaisimmat tietokoneet pystyivät lähettämään 8 bittiä kerrallaan, oli luonnollista aloittaa koodin kirjoittaminen 8-bittisillä numeroilla, jotka merkitsevät yhtä tavua.

Huom! 8 binäärinumerolla voidaan esittää 256 erilaista arvoa. Miksi? Tutki sarakkeita: jos kaikkien sarakkeiden arvona on 1, tulee summaksi 255. Jos sarakkeiden arvot ovat nollia, saadaan summaksi 0. Lukuja (0-255) on siis 256.

Mikä on kilotavu?

2^{10} (1024) on karkeasti ottaen yhtä suuri kuin 10^3 (1000). Tätä yhteensattumaa ei saanut hylätä, joten tietokonemiehet alkoivat viitata lukuun 2^{10} sanalla kilotavu, koska tieteessä etuliite kilo tarkoittaa tuhatta.

Samalla lailla $1024 * 1024$ (1048576) on niin lähellä miljoonaa, että siitä alettiin käyttää sanaa megatavu. 1024 megatavua on yksi gigatavu.

Binäärinumerot

Tietokoneet käyttävät ykkösiä ja nollia tiedon koodaamiseen. Koneen ohjeet ovat sarja ykkösiä ja nollia ja virtapiirit tulkkavat niitä. Epämääräiset ykkösten ja nollien joukot voidaan kääntää takaisin numeroiksi tietokonemiesten toimesta, mutta olisi virhe ajatella näillä numeroilla olevan sisäistä merkitystä.

Esimerkiksi Intelin 80x6-piiri kääntää bittikuvion 10010101 ohjeeksi. Se voidaan tietenkin kääntää takaisin numeroksi, mutta itse numerolla ei ole merkitystä sellaisenaan. Joskus numerot ovat ohjeita, joskus arvoja ja joskus koodeja. Yksi merkittävä standardisoitu koodisto on ASCII-koodi. ASCII-koodissa jokaista merkkiä esittää 7-bittinen binääriluku. Esimerkiksi kirjainta a esittää bittijono 01100001. Se ei ole numero, vaikka sen voi kääntää numeroksi 97. Yleensä on totuttu sanomaan, että kirjainta a esittää luku 97 ASCII-taulukossa, mutta totuus on se, että kirjainta a edustaa bittijono tai koodi 01100001.

Heksadesimaali

Koska binäärilukuja on vaikea lukea, kehitettiin yksinkertaisempi tapa esittää samat arvot. Muuntaminen binäärijärjestelmästä 10-järjestelmään on aikamoista bitteilyä, mutta muuntaminen binäärijärjestelmästä heksamuotoon on melko yksioikoista.

Koeta ensin hahmottaa, millainen on 16-kantainen (heksadesimaali) järjestelmä. Siinä on 16 numeroa: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E ja F. Kirjaimet vastaavat desimaalilukuja 10 - 15 ja niiden käyttö sopii hyvin näppäimistöön. Seuraavassa ovat heksadesimaalin sarakkeet:

| | | | |
|-----------------|-----------------|-----------------|-----------------|
| 4 | 32 | 1 | |
| 16 ³ | 16 ² | 16 ¹ | 16 ⁰ |
| 4096 | 256 | 16 | 1 |

Kääntääksesi luvun heksamuodosta desimaaliin, voit tehdä kertomisen. Siten luku F8C esittää lukua:

$$\begin{array}{rcl}
 F \cdot 256 & = & 15 \cdot 256 = 3840 \\
 8 \cdot 16 & = & 128 \\
 C \cdot 1 & = & 12 \cdot 1 = 12 \\
 \text{Yhteensä} & & 3980
 \end{array}$$

Luvun FC muuntaminen binäärimuotoon kannattaa tehdä muuntamalla luku ensin 10-järjestelmään ja sitten binäärimuotoon:

$$\begin{array}{rcl}
 F \cdot 16 & = & 15 \cdot 16 = 240 \\
 C \cdot 1 & = & 12 \cdot 1 = 12 \\
 \text{Yhteensä} & & 252
 \end{array}$$

Luvun 252¹⁰ muuntaminen binäärimuotoon vaatii taulukon:

| Sarake | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|----------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Potenssi | 2 ⁷ | 2 ⁶ | 2 ⁵ | 2 ⁴ | 2 ³ | 2 ² | 2 ¹ | 2 ⁰ |
| Arvo | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 252 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

Binäärinä luku on siis 11111100.

Jos nyt katsot tuota binäärilukua kahtena 4 bitin ryhmänä, voit tehdä maagisen muuntamisen heksamuotoon.

Oikean puoleinen ryhmä on 1100, joka desimaalilukuna on 12 ja heksana siten C.

Vasemman puoleinen ryhmä on 1111, joka desimaalilukuna on 15 ja heksana siten F.

Saame siis heksaluvun FC.

Tämä on hyvä menettely. Laitat vain bitit 4 bitin ryhmiin ja haet vastaavan heksanumeron. Seuraavassa on pitempi esimerkki:

1011 0001 1101 0111

Sarakkeet ovat:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 ja 32768.

$$1*1=1$$

$$1*2=2$$

$$1*4=4$$

$$0*8=0$$

$$1*16=16$$

$$0*32=0$$

$$1*64=64$$

$$1*128=128$$

$$1*256=256$$

$$0*512=0$$

$$0*1024=0$$

$$0*2048=0$$

$$1*4096=4096$$

$$1*8192=8192$$

$$0*16384=0$$

$$1*32768=32768$$

$$\text{Yhteensä } 45527$$

Tuon muuntaminen heksaksi vaati heksalukutaulukon.

| | | | | |
|-------|------|-----|----|---|
| 65535 | 4096 | 256 | 16 | 1 |
|-------|------|-----|----|---|

65535 ei sisälly kertaakaan lukuun 45527. 4096 sisältyy 11 kertaa lukuun 45527 ja jakojäännös on 471. 256 menee kerran lukuun 471 ja jakojäännös on 215. Luku 16 menee 14 kertaa lukuun 215 ja jakojäännös on 7. Siten heksaluku on B1D7.

Tarkistus:

$$B(11) * 4096 = 45056$$

$$1 * 256 = 256$$

$$D(13) * 160 = 208$$

$$7 * 1 = 7$$

$$\text{Yhteensä } 45527$$

Oikotiemenettelyssä jaamme binääriluvun 1011000111010111 neljän bitin ryhmiin: 1011 0001 1101 0111. Kukin neljän bitin ryhmä muutetaan heksaksi:

$$1011=$$

$$1*1=1$$

$1*2=2$
 $0*4=0$
 $1*8=8$
Yhteensä 11
Heksana B

0001=
 $1*1=1$
 $0*2=0$
 $0*4=0$
 $0*8=0$
Yhteensä 1
Heksana 1

1101=
 $1*1=1$
 $1*2=2$
 $1*4=4$
 $1*8=8$
Yhteensä 13
Heksana D

0111=
 $1*1=1$
 $1*2=2$
 $1*4=4$
 $0*8=0$
Yhteensä 7
Heksana 7

Ja koko luku heksana: B1D7