

## **Viikkoraportti 2**

### **Mitä opin**

Tällä viikolla opin käyttämään gdb:tä ja Valgrindia ohjelmani debuggauksessa. Valgrind on C-ohjelmoinnissa äärimmäisen hyödyllinen, sillä ohjelma voi toimia oikein vaikka se esim. lukisikin arvoja sille varatun muistialueen ulkopuolelta. Microsoftin Visual Studion debuggeriin verrattuna gdb ei ole kovin aloittelijaystävällinen, mutta pienen perehtymisen jälkeen peruskäyttö ei enää ollut mikään ongelma. Tekstipohjainen käyttöliittymä mahdollistaa mm. monipuolisen skriptauksen, mihin Microsoftin työkalu ei pysty.

### **Mikä jäi epäselväksi**

Yritin tällä viikolla ottaa selvää C-ohjelmien oikeaoppisesta rajapintasuunnittelusta, mutta en ole varma oliko tästä mitään apua. Ilmeisesti järkevin tapa muodostaa abstrakti tietorakenne (joka siis toimii millä tahansa tietotyyppillä) on suunnitella se käyttämään ns. läpinäkymättömiä osoittimia. Toisin ilmaistuna tietorakenteen implementaation ei tule saada tietää lopullisesta tietotyypestä oikeastaan mitään, ja tämä yleensä saadaan aikaan luovalla `typedef`-avainsanan käytöllä. Tulen tutustumaan asiaan vielä lisää kun toteutan Huffman-koodauksen vaatiman puurakenteen.

### **Ohjelman edistyminen**

Tällä viikolla lisäsin ohjelmaan toiminnot kuvien manipuloimiseksi. PPM-muotoon tallennettuja kuvia pystyy nyt avaamaan, jolloin ne muutetaan ohjelman sisäiseen bittikarttaformaattiin. Myös kuvan paloittelu DCT:n tarvitsemaan 8x8-ruudukkoon toimii. Myös funktio bittikarttojen tallentamiseen levyille (samaiseen PPM-formaattiin) on nyt implementoitu. Kaikki lisätyt ominaisuudet on myös testattu. Pienten virheiden etsimiseen meni kauemmin aikaa kuin odotin, joten en ehtinyt toteuttaa kaikkea mitä viime viikolla uumoilin.

### **Mitä seuraavaksi**

Seuraavaksi ladatut kuvat pitäisi saada vielä muutettua taajuusmuotoon. Tämä on helppoa, sillä tämä osio ohjelmasta on jo käytännössä toteutettu, yksittäisten moduulien toiminnot pitää vain yhdistää keskenään. Tämän jälkeen pitäisi toteuttaa taajuuskertoimien uudelleenjärjestely taulukossa, ja tämän lopputuloksen RLE-pakkaus. Sitten jäljellä olisi enää lopullisen binääridatan Huffman-koodauksen toteutus, jonka voi todennäköisesti eristää täysin omaksi komponenttikseen.