Eemil Aspholm

CT30A3370 Käyttöjärjestelmät ja systeemiohjelmointi

Viikkotehtävä 4.

**Tehtävä 1.**

**./lottery.py -s 1 -j 3 -c**

**./lottery.py -s 2 -j 3 -c**

**./lottery.py -s 3 -j 3 -c**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

2.

**./lottery.py -l 10:1,10:100 -c**

Kun ohjelmaa suoritetaan tällaisella epätasapainolla, on valittava työn numero 1 100 lippua yhteensä 101 lipusta kymmenen iteraatiota peräkkäin. Tästä johtuen työllä numero 0 on 1/(1+100) mahdollisuus eli n. 1% mahdollisuus tulla suoritetuksi ennen tehtävän 1 valmistumista. Prosentuaalisen todennäköisyyden vuoksi voitaisiin sanoa, että tehtävä 1 valmistuu melkeinpä aina ennen tehtävän 0 valmistumista. Epätasapaino vaikuttaa olennaisesti töiden valmistumisjärjestykseen.

3.

Kokeilujen perusteella, kun kumpienkin töiden pituus ja tikettien jako on samansuuruinen, on jakautuminen melko reilua. Testien perusteella voidaan myös nähdä, että mitä pidempi työ on, sitä suurempi on myös oikeudenmukaisuuden todennäköisyys.

**./lottery.py -s 0 -l 100:100,100:100 -c**

**192/200 = 0.96**

**./lottery.py -s 1 -l 100:100,100:100 -c**

**196/200 = 0.98**

**./lottery.py -s 2 -l 100:100,100:100 -c**

**190/200 = 0.95**

**./lottery.py -s 3 -l 100:100,100:100 -c**

**196/200 = 0.98**

4.

Kvanttikoon muuttuessa isommaksi voidaan huomata samanlainen ilmiö kuin työn pituuden pienentyessä -> epätasapaino / epäluotettavuus kasvaa isommaksi.

Table

Description automatically generated with medium confidence

**Tehtävä 2.**

1. Pythonissa .py päätteiset tiedostot ovat python-koodia jonka ulkoasu on ihmiselle perinteikkäässä ja luettavassa muodossa. Esim ( if(x=1) do y = 2x). Pythonin .pyc tiedostot ovat puolestaan tavukoodia, jota ihminen ei pysty lukemaan. Python ohjelmia suoritettaessa python kääntää .py syntaksisen ohjelmatiedoston tavukoodiksi .pyc tiedostoksi, jota se lopulta sitten käyttää tulkintaansa ohjelman suorittamiseksi. Lisäksi jos ohjelmassa on esimerkiksi import käskyllä sisällytettyjä moduuleja ohjelmaan, tällöin pyhon kääntää moduulin sisältämän koodin ja luo .pyc tiedoston, tämä helpottaa ohjelmistokoodin suorittamista, sillä suorituksen yhteydessä moduuleita, joihin ei ole tullut muutoksia ei tarvitse kääntää uudestaan tavukoodiksi.
2. Python ohjelman tavukoodi näyttää omasta mielestäni helpompi tulkinnalliselta verrattuna viime viikkoisen assembly tehtävän koodiin. Pystyn yhdistämään tavukoodista elementtejä .py koodin syntaksiin kuten funktion nimen ”foo”. Tavukoodi itsessään sisältää **.co\_code** syntaksin jolla tuodaan esiin että ”foo” on ohjelman funktio/metodi. **[byte for byte in foo .\_\_code\_\_.co\_code]** syntaksi puolestaan kertoo että funktiossa foo. On luettavia tavuja, ts merkkijono. Mielestäni ohjelmistokoodista tulisi huomattavasti helpompi tulkintaista, mikäli turha toisto / \_co\_code jätettäisiin pois perästä.
3. Kuten aiemmin mainittu pythonin tavukoodi omasta mielestäni on huomattavasti loogisempaa ja luettavampaa verrattuna assemblyn syntaksiin. Python ohjelmointikieli ja assembly ovat ensinnäkin erilaisia ohjelmointikieliä, python kuten aiemmin mainittu on tavukoodia, ja asembly puolestaan konekieltä. Pythonin tavukielessä muuttujat ja vakioarvoja tallennetaan pinoon.
4. POP\_TOP poistaa pinon ensimmäisen jäsenen, joka käytännössä tarkoittaa, että pinon päällimmäinen operandi on suoritettu.

**Tehtävä 3.**

**Table

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence**

Yläpuolelle liitettyjen funktioiden tavukoodit ovat identtiset, eivätkä ne eroa, vaikka ohjelmistokoodin syntaksi eroaa else if /elif rakeenteella. Voidaan siis tehdä johtopäätös, että elif/else if syntaksin tavukoodi on identtinen kuten myös niiden toiminnallisuus. Yritin myös vaihtaa else if osioiden järjestystä, mutta en saanut silläkään muutoksia tavukoodin syntaksiin. En ole aivan varma teinkö oikeanlaisia muutoksia, sillä tehtävän annon kysymys oli hieman epäselvä. Mikäli .py tiedostossa muutetaan else if rakenteen sijaintia, muuttuu myös tavukoodi tiedoston syntaksin sijainti.