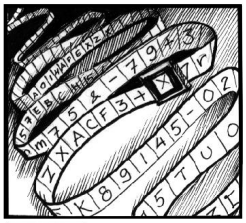

Mitä tietokonevirheistä seuraa? Räjähdyksiä, uppoamisia ja tappavaa säteilyä

Juha Haataja

Tietotekniikan ongelmat aiheuttavat mitä ihmeellisimpiä virhetilanteita. Usein virheen syyksi paljastuu väärin toimiva ohjelmakoodi. Syynä voi olla triviaali virhe ohjelmakoodissa tai syvällinen suunnitteluvirhe ohjelmiston rakenteessa.

Klassinen esimerkki ohjelmistovirheestä on "Wednesday-koodi", joka toimi vain keskiviikkoisin. Tämä johtui siitä, että keskiviikon nimessä oleva y-kirjain kirjoitettiin seuraavan kentän päälle ja tämä y-merkki sai koodin toimimaan oikein (y = yes).

Miten luotettavia tietokoneohjelmistot ovat? Käytännön kokemus osoittaa, etteivät kovinkaan. Tilannetta kuvaamaan onkin syntynyt termi "banaaniohjelmistot": käyttäjät kypsyttävät raa'at tuotteet jotakuinkin käyttökelpoisiksi.

Esittelen seuraavassa muutamia tieteeseen ja tekniikkaan liittyviä tapahtumia, joissa tietoteknisillä virheillä on ollut merkittävä rooli. Tietotekniikan käyttö ei ole tietenkään pelkkiä virheitä ja ongelmia. Parhaimmillaan tietotekniikka on erinomainen työkalu ja apuväline. Tuon esille ongelmatilanteita eri tyyppisten virheiden havainnollistamiseksi. Ehkä kokemuksesta voi oppia.

Urbaaneja legendoja

Vuonna 1962 Nasa laukaisi Venukseen tarkoitetun Mariner I -luotaimen. Pian lähdön jälkeen raketti alkoi käyttäytyä holtittomasti, minkä takia se jouduttiin tuhoamaan. Luotain putosi Atlantin valtamereen.

Tapaturman syynä oli laitevirhe yhdistyneenä ohjelmistovirheeseen. Laitevian takia rakettia ohjattiin tutkan avulla maasta käsin. Tutkan antamissa mittaustiedoissa oli virhettä, minkä takia mittausarvoista olisi pitänyt laskea juokseva keskiarvo.

Ohjauskoodin suunnitelmista kuitenkin puuttui keskiarvostusta tarkoittava yläviiva nopeusmuuttujan päältä. Siten ohjaukseen käytetiin viimeisintä tutkan antamaa arvoa, jossa oli mukana satunnaista virhettä. Ohjausjärjestelmä kuvitteli raketin heittelehtivän ja yritti kompensoida tätä komennoilla, jotka todella saivat raketin käyttäytymään holtittomasti.

Mariner I -luotaimen tuhoutumisesta muodostui ydin monille tarinoille. Koska ongelman syy oli hiukan monimutkainen, puhutaan tarinoissa yleensä koodissa olleesta etumerkkivirheestä tai pilkkuvirheestä. Pilkkuvirhetarinan alkuperä löytyy ilmeisesti seuraavassa kuvatusta ongelmasta, joka tapahtui samoihin aikoihin.

Matkalla avaruuteen

Vuonna 1963 Nasassa kehitettiin ja testattiin aiemmilla Mercury-lennoilla käytettyä rakettisimulaattoria. Testauksessa havaittiin, että tulokset olivat kohtalaisen tarkkoja, mutta eivät kuitenkaan täysin vastanneet tunnettuja tuloksia. Usean viikon testauksen jälkeen Fortran-ohjelmakoodista löytyi rivi

```
DO 10 I=1.10
```

Tässä piti olla toistorakenne, jota suoritetaan kymmenen kertaa. Pilkun vaihtuminen pisteeksi muunsi lauseen kuitenkin sijoituslauseeksi, jossa muuttujaan `DO10I` sijoitettiin arvo `1.10`. Siispä kyseinen toistorakenne suoritettiin vain kerran. Saadut tulokset olivat riittävän tarkkoja aiemmilla lennoilla, jolloin rakettien tehot olivat pienempiä. Onneksi virheestä ei aiheutunut mitään todellisia vahinkoja. Nykyisessä Fortran-kielen versiossa tämänkaltaiset virheet huomattaisiin jo ohjelman käännösaikana.

Euroopassa kehitetyn Ariane 5 -raketin ensimmäinen laukaisu tapahtui 4.6.1996. Rakettia oli kehitetty vuosikymmenen ajan ja kehityskulut olivat luokkaa 50 miljardia markkaa. Noin 37 sekuntia laukaisun jälkeen raketti alkoi käyttäytyä holtittomasti ja lopulta räjähti. Raketin ja sen lastin arvo oli useita miljardeja markkoja.

Turman syy selvisi kahden viikon kuluessa. Ohjelmassa käytettiin Ariane 4 -raketille kehitettyä ohjauskoodia, jossa raketin vaakasuoraa nopeuttaa kuvaava 64-bittinen liukuluku muutettiin 16-bittiseksi kokonaisluvuksi. Tässä tapauksessa lukuarvo oli kuitenkin yli 32 768, joka on suurin 16-bittisessä kokonaislukuaritmetiikassa esitettävissä oleva luku. Prosessori antoi virhetilanteesta ilmoituksen ja tulosti virheraportin.

Virhetilanteen käsittelyä ei kuitenkaan määritelty Adakoodissa, jolloin ohjausjärjestelmä yritti tulkita tuloksen raketin ohjauskomennoiksi. Seurauksena oli holtiton käyttäytyminen ja lopulta raketin itsetuhomekanismin käynnistyminen. Kyseinen osa ohjauskoodia ei ollut tarpeen Ariane 5:ssä ja oli joka tapauksessa ohjelmoitu poistumaan käytöstä 40 sekuntia laukaisun jälkeen.

Nasan Marsiin lähettämä Climate Orbiter -luotain on viimeisimpiä avaruusmatkailun takaiskuja. Luotain tuhoutui 23.9.1999 syöksyttyään Marsiin. Virheen syyksi selvisi mittayksikkövirhe ohjausrakettien tehon määrittelyssä. Lockheed Martin oli käyttänyt määrittelyissä englantilaisia yksiköitä ja Nasan Jet Propulsion Laboratory puolestaan oletti käytettävän metrijärjestelmän yksiköitä (paunat vs. Newtonit). Tämän johdosta raketti ajautui 80 kilometriä ohi kurssin ja törmäsi Marsiin.

Indeksit matkalla etelään

Vuonna 1982 Vancouverin pörssi otti käyttöön uuden pörssi-indeksin, jota päivitettiin jokaisen kaupan jälkeen. Indeksin alkuarvoksi asetettiin 1000. Indeksin arvo putosi 20 kuukauden kuluessa 520:een.

Syynä oli laskennassa käytetty katkaiseva aritmetiikka: päivitettyä indeksin arvoa ei pyöristetty lähimpään tuhannesosaan vaan arvo katkaistiin ja loput desimaalit unohdettiin. Pyöristystä käyttäen saatiin indeksin arvoksi 1099.

Tappavaa säteilyä

Vuosina 1985-87 aiheutui säteilyhoitoon käytetyn Therac-25 -laitteiston toimintavirheestä useita kuolemantapauksia ja loukkaantumisia. Laitteisto edusti uutta tekniikkaa ja oli aiemmista versioista poiketen kokonaan tietokoneohjattu. Turvallisuuden varmistaminen oli hoidettu ohjelmallisesti aiempien laitemekanismien sijaan. Riskianalyysissä unohdettiin huomioida mahdollisten ohjelmistovikojen vaikutukset toimintaan.

Vuosina 1985-87 tapahtui kuusi massiivista säteilyn yliannostusta potilaille. Säteilymäärät olivat pahimmillaan jopa yli satakertaisia normaaliin säteilyhoitoon verrattuna.

Ongelman syyksi osoittautui laiteoperaattorin käyttöliittymä: jos käyttäjä editoi koneelle annettavia komentoja liian nopeasti, hyväksyi kone virheellisen annostusmääräyksen. Koska annostusmääräystä ei tarkistettu eikä koneessa ollut yliannostuksen havaitsevia sensoreita, ei virhetilannetta havaittu ennen kuin potilaat valittivat säteilyn aiheuttamista akuuteista oireista. Säteilyn yliannostuksesta oli seurauksena ainakin kolme kuolemantapausta.

Ohjuksia väärään kohteeseen

Vuonna 1988 USA:n hävittäjäkone ampui alas Iran Air -lehtoyhtiön Airbus A300B2 -koneen. Lento 655 lähti Iranista Bandar Abbasin lentokentältä ja oli matkalla Dubaihin. Persianlahdella ollut risteilijä USS Vincennes havaitsi lennon Aegis-lennonvalvontajärjestelmässään. Vaikka lento oli jokaviikkoinen, ei sitä löydetty vakiolentojen aikataulusta.

Tietokonejärjestelmä oletti koneen olevan F-14 -hävittäjä, joten risteilijästä lähettiin pyyntö iranilaiselle F-14 -hävittäjälle tunnistautua. Tällöin matkustajalentokone keskusteli yhä lennonjohdon kanssa.

Vahvistus lentokoneen vihamielisistä aikeista saatiin, kun Aegis-järjestelmä näytti ilmoittavan koneen olevan nopeassa syöksyssä normaalien lentoreittien ulkopuolella kohti Vincennessiä. Todellisuudessa kone oli yhä nousussa ja normaalilla lentoreitillä. Risteilijästä annettiin käsky ampua lentokone alas. Koneessa kuoli 290 ihmistä.

Vuoden 1991 alussa USA ja Irak kävivät sotaa Persianlahdella. Irak ampui Scud-ohjuksia amerikkalaisiin sotilaskohteisiin ja USA käytti torjuntaan Patriot-ilmatorjuntaohjuksia. Kuitenkaan 25. helmikuuta Patriot-ohjus ei osunut kohteeseensa ja Scud-ohjus tappoi 28 amerikkalaissotilasta.

Syyksi osoittautui ohjelmistovika. Patriot-ohjuksessa on kello, joka mittaa ajan kulumista kymmenesosasekunteina käyttäen kokonaislukulaskuria. Ohjusjärjestelmä oli ollut yhtäjaksoisesti toiminnassa yli 100 tuntia. Siis laskurin arvo oli suuruusluokkaa 3,6 miljoonaa.

Patriot-ohjus etsii tulevaa ohjusta alueelta, jonka paikka arvioidaan edellisen mittausarvon perusteella. Kulunut aika määrätään kertomalla aikalaskurin arvo luvulla 0,1. Koska luvun 0,1 binääriesitys katkaistiin 24 bittiin, oli tämän luvun esitysmuodossa suuruusluokkaa 10^{-7} oleva virhe. Tämä luku kerrottiin luvulla $3{,}6 \cdot 10^6$, joten tulokseen tuli virhettä noin 0,3 sekunnin verran. Tässä ajassa Scud-ohjus ehti lentää yli 600 metriä, joten Patriot-ohjus yritti paikantaa tulevaa ohjusta aivan väärältä suunnalta.

Pelataan lautanupotusta

Sleipner A -öljynporauslautta tuottaa öljyä ja kaasua Pohjanlahdella 82 metrin syvyisessä vedessä. Lautta on rakennettu betoniselle alustalle. Alustasta kohoaa neljä tornia, joiden varassa on lautan laitteistokansi.

Lautan alustaa testattiin painolastin avulla 23.8.1991 ennen kannen asennusta paikalleen. Testauksessa alustaan tuli vuoto ja se upposi vuonoon Stavangerin lähellä. Uppoaminen 220 metrin syvyyteen aiheutti järistyksen, jonka suuruus oli 3,0 Richterin asteikolla. Taloudelliset tappiot olivat miljardiluokkaa.

Tutkimuksissa kävi ilmi, että yhteen alustan seinämistä tuli vakava vuoto, jota pumput eivät pystyneet kompensoimaan. Syynä oli suunnittelu- ja rakennusvirhe. Alustan lujuuslaskelmat oli tehty elementtimenetelmällä käyttäen NASTRAN-ohjelmistoa. Alustan osasten liitoskohdan analyysissä oli käytetty vääränlaista elementtimallia, jolloin osaan vaikuttavia voimia aliarvioitiin lähes 50%. Tarkemmissa laskelmissa päädyttiin tulokseen, että rakenteen kestävyys pettäisi 62 metrin syvyydessä. Todellisuudessa rakenne petti 65 metrin syvyydessä.

Virheellistä laskuoppia

Vuonna 1994 havaittiin virhe Pentium-prosessorin jakolaskuoperaation tuloksissa. Virhe esiintyi harvoin, mutta oli potentiaalisesti merkittävä. Virheen suhteellinen suuruusluokka oli pahimmillaan 10^{-5} ja tuloksessa oli tarkkuutta vain 14 bittiä. (Tätä voi verrata Patriot-ohjuksen aritmetiikkavirheeseen.) Intelin maine koki virheen ansiosta pahan takaiskun. Lopulta Intel lupasi vaihtaa virheelliset prosessorit virheettömiin.

Vuonna 1998 Intelin Pentium II ja Pentium Pro -prosessoreissa havaittiin virhe ylivuototilanteen käsittelyssä. Jos liian iso liukuluku yritettiin muuntaa 16-bittiseksi kokonaisluvuksi, prosessorin olisi pitänyt antaa tilanteesta virheilmoitus. Kuitenkaan prosessori ei tehnyt tätä kaikissa tilanteissa, joten virhe jäi havaitsematta. Tätä virhettä voi verrata Ariane 5 -raketin aritmetiikkavirheeseen.

Ohjelmistojen luotettavuus

Tieteen ja tekniikan ohjelmistojen luotettavuutta on selvitetty useissa tutkimuksissa. Eräs perusteellisimmista oli lehdessä *IEEE Computational Science & Engineering* (April−June 1997) esitelty vertailu.

Lehdessä oli tutkittu FORTRAN 66/77 ja C-kielisiä tieteen ja tekniikan ohjelmistoja. Testi koostui kahdesta vaiheesta: ohjelmakoodien staattisesta analyysistä sekä seismisten analyysiohjelmistojen vertailusta. Lähdekoodin staattisessa analyysissä oli tutkittavana 55 FORTRAN-ohjelmistoa ja 68 C-kielistä ohjelmistoa, joissa oli yhteensä 3,3 miljoonaa FORTRAN-kielistä koodiriviä ja 1,9 miljoonaa C-kielistä koodiriviä. Eri sovellusalueita oli 40.

Suurin osa koodeista oli peräisin kaupallisista yrityksistä ja kaikki koodit olivat tuotantokäytössä. Koodien käyttäjät uskoivat koodien olleen täysin testattuja.

FORTRAN-koodeissa oli keskimäärin 12 vakavaa virhettä 1000 koodiriviä kohden; C-koodeissa puolestaan oli 8 vakavaa virhettä 1000 riviä kohden. Eräässä ydintekniikan koodissa oli 140 virhettä 1000 koodiriviä kohden. Tämä koodi onkin lähinnä hyvin kallis satunnaislukugeneraattori.

Proseduurien kutsut olivat yhteensopimattomia joka 7. tapauksessa FORTRAN-koodeissa ja joka 37. tapauksessa C-koodeissa. Ero johtunee lähinnä FORTRAN-koodien suuremmasta argumenttien lukumäärästä sekä automaattisten tarkistusten puutteesta. (Nykyisessä Fortran 95 -standardissa on kehittyneempiä virheentarkistuksia.)

Osa koodeista oli kirjoitettu käyttäen hyvin hämärää ja virhealtista ohjelmointityyliä. Pahimmassa esimerkissä oli 500 000 000 erilaista reittiä ohjelmayksikön läpi. Pienikin muutos tällaiseen koodiin saattaa muuttaa koodin käyttäytymisen täydellisesti. Siten kyseisen koodin ylläpidettävyys on olematon.

Ohjelmistovertailussa tutkittiin seismistä dataa käsitteleviä ohjelmistoja. Seismistä analyysiä käytetään maaperän rakenteen selvittämiseen, jotta voidaan valita oikea paikka koeporauksille. Yksi poraus voi maksaa kymmeniä miljoonia markkoja, joten tulosten pitäisi olla luotettavia.

Testattavina oli yhdeksän toisistaan riippumattomasti kehitettyä tuotetta. Seismisen datan analyysissä käytetty matemaattinen algoritmi on suhteellisen yksinkertainen ja käytössä kaikissa testatuissa ohjelmistoissa. Testissä annettiin kaikille ohjelmistoille sama syöttödata, jonka jälkeen tuloksia verrattiin sekä koodien kesken että ajamalla samaa koodia eri koneissa ja eri tarkkuuksilla.

Useista koodeista löytyi tyypillisiä "yhdellä pielessä" indeksointivirheitä datan analyysissä. Toisaalta samalla ohjelmistolla eri koneissa ja eri tarkkuuksilla saadut tulokset olivat muutaman desimaalin tarkkuudella identtiset.

Ikävä kyllä ohjelmistojen keskinäinen vertailu paljasti, että saadut tulokset olivat erilaiset: tuloksissa oli yhteneväisyyttä noin yhden merkitsevän numeron verran. Lisäksi osa koodeista oli ilmeisen virhealttiita: laskennan kuluessa saadut tulokset erosivat yhä enemmän "keskimääräistästä tuloksista". Yksikään koodeista ei näyttänyt olevan hyvä kaikissa vertailupisteissä: kullakin tuntui olevan sokeat pisteensä. Yksi koodeista tosin oli johdonmukaisen huono, mutta muut kilpailivat menestyksellisesti huonouden kakkossijasta.

Tämä artikkeli on julkaistu lähes samassa muodossa *Tietoyhteys*-lehden numerossa 1/2001, ja se julkaistaan Solmussa *Tietoyhteys*-lehden luvalla.