

Vaarallisia ohjelmia

ITKA203 Käyttöjärjestelmät -kurssin Demo 3 kesällä 2007. Paavo Nieminen / Jyväskylän yliopiston Tietotekniikan laitos.

“Kokeillaan käytännössä käyttöjärjestelmän kuormittamista”

Lue ohjeet läpi ennen kuin aloitat!

1 Mistä tässä harjoitteessa on kyse

Materiaali syntyi Käyttöjärjestelmät -kesäkurssilla 2007. Kiitän kurssilaisia palautteesta. Tähän käyttivät aikaa noin puolitoista tuntia. Oli kuulemma ihan valaisevaa. Tahtoisin kuvitella, että puhuivat totta.

Harjoituksen tavoitteet ovat vähintään ainakin seuraavat:

- Harjaannut lukemaan lyhyitä C-ohjelmia
- Näet miten käyttöjärjestelmäkutsuja tehdään C-kielestä käsin
- Tutustut käytännön ohjelmakoodin kautta mm. `fork()` -käyttöjärjestelmäkutsuun sekä tapaan, jolla uusi prosessi luodaan Unix-ympäristössä.
- Tutustut käytännössä kuormituskokeilujen kautta käyttöjärjestelmään ja prosesseihin
- Tutustut `top` -järjestelmäseurantaohjelmaan
- Näet seurauksineen asioita, joita sinun ei ehkä koskaan kannata oikeissa töissä tehdä, ainakaan vahingossa ja tietämättäsi.

2 Ohjeet

Toimi mikroluokassa AgC334.1 Linux-koneella! Luokan käytössä ota huomioon seuraavat asiat:

- luokka on avoinna virastoaikaan eli ainakin klo 10-17.
- jos näyttö ja näppäimistö on Windows-koneen puolella, vaihda Linuxiin: paina kaksi kertaa peräkkäin “Scroll Lock” -näppäintä.
- Lopussa kirjoittaudu ulos kaikista koneista, joihin olit kirjautunut (ja myös Linuxin tekstimuotoisista konsoleista, jos käytit niitä). X-ikkunoidusta työpöydästä löytynee “Desktop” -valikosta valinta “Log Off” tms.
- Mikroluokassa AgC334.1 toimii samat käyttäjätunnukset kuin muuallakin, mutta Linux-työasemat on eristetty muusta maailmasta esim. siten, että verkkolevy (“U-asema”) ei näy koneissa, vaan niissä kotihakemisto on luokan AgC334.1 sisäisellä pikkuverkkolevyllä.
- Jos koneet tässä luokassa menevät jumiin, niin ei kuulemma haittaa vaikka resetoit ne, tai katkaiset hetkeksi virran: Muita käyttäjiä ei pitäisi olla verkkoyhteyden kautta, vaan olet yksin kyseisen koneen ainoa käyttäjä. Koneilla ei pitäisi olla mitään arvokasta dataa, jonka tuhoutuminen haittaisi ketään. Myös käyttöjärjestelmän korruptoituminen lienee näissä helposti korjattavissa jollain tavoin. Kyseessä on siis “idioottivarma mikroluokka”, varsinaisesti tietoliikenteen puolen projektitöihin. Täällä saa leikkiä sellaista, mitä muualla ei. Siis tehdään nyt juuri niin ...
- Terminaali-ikkunan saa päälle klikkaamalla oikeata nappia työpöydän päällä ja valitsemalla “Open terminal”. Siihen sitten käynnistyy interaktiivinen shell, josta paikallista konetta voi komentaa.
- X-ikkunoidussa työpöytä-linuxissa voit käyttää graafisia editoreja, kuten gedit. Esim. Valikosta “Applications”->“Run Application” ja kirjoita boksiin `gedit`. Käännökset ym. tehdään kuitenkin terminaali-ikkunassa (joka näyttää ihan samalta kuin pääteyhteys interaktiiviseen shelliin etäkoneella; tällainenhan tuli tutuksi demossa 1)
Terminaalin shellistä saa myös päälle editorit ja muut ohjelmat. Niistä tulee kyseisen shellin lapsiprosesseja, joten ne lopetetaan, jos shell suljetaan väkisin. Ne saa tausta-ajoksi suoraan kirjoittamalla komennon perään `Et-merkki eli &` (Esimerkiksi voisit komentaa shellistä `gedit &`. Shell jää silloin odottamaan seuraavaa komentoa, mutta käynnistetty ohjelma jää taustalle ihan normaalisti suoritukseen. (Ja vaikka se on taustalla, se on edelleen shellin lapsiprosessi).
- Nettiselain Firefox lähtee päälle siitä kuvakkeesta, jossa on maapallon ympärillä hiiri tai jotain sinne päin... (tai komentamalla shellistä se tausta-ajoksi: `firefox &`)

Tee AgC334.1:n Linux-koneella joku hakemisto tätä demoa varten ja mene sinne (terminaali-ikkunassa; `mkdir` tekee hakemiston ja `cd` vaihtaa shellin työhakemistoa)

Shellin käyttö tuli tutuksi demossa 1. Kertaa sieltä, jos et muista miten joku juttu toimi.

Hae <http://www.cc.jyu.fi/~nieminen/kj07kesa/demo3.zip>

Avaa paketti.

Käy läpi ohjelmat, mieluiten seuraavassa järjestyksessä:

```
forkkaa_loppuun_asti1.c
forkkaa_loppuun_asti2.c
forkkaa_loppuun_asti3.c
forkkaa_loppuun_asti4.c
muistivuoto.c
muistivuoto_korjattu.c
skripti_idiootti.sh
```

Tutustu jokaiseen, koeta ymmärtää kaikki se mitä ohjelmissa tapahtuu. AgC334.1:ssä saat kokeilla niitä mielin määrin, eli käänös gcc:llä ja ajo terminaali-ikkunassa.

Käynnistä lisäksi toinen terminaali-ikkuna `top` -ohjelman käyttöä varten. `Top` kertoo reaaliaikaisen tilanteen käyttöjärjestelmän prosesseista: Montako on luotu, moniko on missäkin suoritustilassa, paljonko fyysistä ja virtuaalista muistia on käytettävissä ja paljonko siitä on vapaana. Näiden lukuarvojen seuraaminen valaisee kunkin ohjelman toimintaa ja vaikutusta käyttöjärjestelmään. Tietenkin prosessori suorittaa kerrallaan vain yhden prosessin käskyjonoa, eikä mikään muu prosessi ole Running-tilassa silloin kun `top` on. Niinpä `top` näyttää tilastotietoa viimeksi kuluneelta hetkeltä (muistaakseni jollain `topin` argumentilla voi säätää tuon tarkasteluhetken pituutta, mutta defaultti tuntuu riittävän tähän tarkoitukseen ihan hyvin).

Esimerkiksi seuraaviin kannattaa kiinnittää huomiota:

- Missä vaiheessa systeemi alkaa “tökkimään”?
- Mitä se silloin tekee?
- Mistä tökkiminen siis luultavasti aiheutuu?
- Miten vuorontaja kohtelee prosesseja, joilla laskenta alkaa kestää?

3 Palautustehtävä

Tämän kerran palautustehtävä on annettu skriptin `skripti_idiootti.sh` kommenttina. Skriptiäkin kokeillessa voinet varmistua asioiden toiminnasta käyttämällä `top`-ohjelmaa toisessa ikkunassa.

(Tehtävä on vapaaehtoinen; annan siitä yhden pisteen bonusta ensimmäiseen tenttiyrytykseen, jos palautus on tehty ennen kesän 2007 ensimmäistä tenttipäivää.)