

Harjoitustehtäväkierros 3

Kurssiassistentit
t-110.4100@tkk.fi

Kevät 2012

Protokollapinon toteutus

Kolmannella tehtäväkierroksella tutustutaan oikeaan TCP/IP-toteutukseen. Tehtävänä on selvittää, kuinka Contiki-käyttöjärjestelmästä löytyvä uIP-pino toimii. Ensimmäisessä tehtävässä toteutetaan Contikille pieni asiakasohjelma. Toisessa tehtävässä tutkitaan uIP-pinon lähdekoodia.

Bonustehtävässä muokataan itse pinon toiminnallisuutta. Tämä tehtävä ei ole kurssin suorittamisen kannalta pakollinen, mutta sillä voi hankkia maks. kaksi lisäpistettä kurssin harjoitusarvosanaa varten.

Tehtävä 1

Tässä tehtävässä tutustutaan Contiki-käyttöjärjestelmään. Contiki on su-lautettuihin järjestelmiin suunniteltu käyttöjärjestelmä, joka sisältää minimalistisen uIP-protokollapinon. Tämä protokollapino mahdollistaa tutustumisen sen toteutukseen kurssin puitteissa, yleisimpien käyttöjärjestelmien protokollapinot voivat olla massiivisia ja vaikeasti lähestyttäviä.

- A Contikin ydin on ns. tapahtumapohjainen. Minkätyyppinen ydin omassa koneessasi on? Miten nämä eri mallit eroavat toisistaan?
- B Mitä ovat TUN ja TAP?
- C Mitä ovat protosocketit ja miten niitä käytetään? Mitä muuta tapaa voit käyttää tcp-yhteyksillä viestimiseen uIP:llä?

Ohjelmointitehtävä:

Hae ja pura paketti `contiki.zip` (löytyy Nopasta). Muokkaa Contiki-käyttöjärjestelmälle kirjoitettua ohjelmaa `assignment/assignment.c` niin,

että pystyt antamaan sillä käskyjä palvelimelle ja ohjelma tulostaa vastauksena tulleen viestin (tyyliin telnet). Voit käyttää `netcat`-ohjelmaa koneellasi testauksen apuna.

Koko projekti käännetään antamalla komento `make assignment`-hakemistossa, jonka jälkeen voit ajaa koko käyttöjärjestelmää oman koneesi käyttöjärjestelmän päällä. Verkkoasetuksistasi riippuen voit joutua säätämään virtuaalisen verkkorajapinnan asetuksia tiedostosta `platform/minimal-net/contiki-main.c`. Liitä palautukseesi muokattu lähdekooditiedosto `assignment.c`.

HUOM: Tarvitset riittävästi oikeuksia, jotta voit muodostaa käyttöjärjestelmään virtuaalirajapinnan.

Tehtävä 2

- A Ensimmäisessä tehtävässä käytettiin protosocket-rajapintaa tiedon lähettämiseen ja vastaanottamiseen. Selvitä (ja kerro!), miten tieto liikkuu `PSOCK_SEND(psock, data, datalen)` -makron seurauksena uIP-pinossa kuljetuskerrokselta ulos.
- B Miten uIP-pinon toteutus eroaa puhtaasta protokollapinomallista ja miksi tällaiseen ratkaisuun on päädytty?

OHJEISTUSTA: Tehtävää varten kannattaa tutustua ainakin seuraaviin:

- Contikin dokumentaatio luvut 6.7 ja 6.49 (pdf mukana paketissa)
- `/core/net/psock.*`
- `/core/net/uip.*`

Bonustehtävä

Muokkaa uIP-pinon toteutusta niin, että TCP-paketteja lähetetään Go-Back-N -periaatteella. Kerro dokumentissa miten ratkaisusi toimii ja palauta muokkaamasi lähdekooditiedostot.

OHJEISTUSTA: Aloita pohtimalla mitä nykyiseen toteutukseen pitää lisätä, jotta olisi mahdollista lähettää useita paketteja ilman, että kuittauksia odotellaan jokaiselle erikseen. Tältä pohjalta voit suunnitella tarvittavat muutokset. Sinun ei myöskään tarvitse ottaa huomioon laitteiden vähäisiä resursseja, sillä et ole nyt sulautettujen järjestelmien kurssilla. Nyt tarkoituksena on tutustua protokollapinon toteuttamiseen.