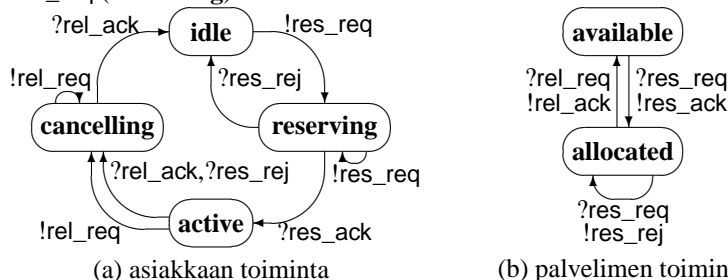


Hajautetun algoritmin toteuttaminen

Keskinäisen poissulkevuuden eli resurssinvarauksen ongelma voidaan ratkaista monella tavalla. Seuraava ratkaisu perustuu "puheenjohtajaan", asiakkaille vuoroja jakavaan palvelimeen, johon asiakkaat ovat yhteydessä häviöllisen kanavan välityksellä.

Asiakkaat lähettävät kahdenlaisia viestejä: varauspyyntöjä `res_req` ja vapautuspyyntöjä `rel_req`. Vapautuspyynnöt palvelin kuittaa aina viestillä `rel_ack`, mutta varauspyyntöön se vastaa joko myöntävästi (`res_ack`) tai kieltävästi (`res_rej`).

Palvelin voidaan esittää tilakoneena, jolla on $n + 1$ tilaa: joko resurssi on vapaa (alkutila **available**) tai se on varattu jollekin n asiakkaasta (**allocated**). Asiakas on nelitilainen tilakone, alussa jouten (**idle**). Se voi odottaa kuittausta varauspyyntöönstään `res_req` (**reserving**), pitää hallussaan resurssia (**active**) tai odottaa kuittausta vapautuspyyntöönstään `rel_req` (**cancelling**):



Tilakone voidaan esittää myös taulukkona, jonka sarakkeista katsotaan nykyinen tila ja riveiltä saapuneen viestin nimi. Vastaavassa solussa lukee uuden tilan nimi ja mahdollisesti lähetettävän viestin nimi. Viiva tarkoittaa, ettei tila muutu. Asiakas lähettää pyyntöjä `res_req` ja `rel_req` joko aikavalvonnan lauetessa tai saadessaan käyttäjältään pyynnön varata tai vapauttaa resurssi.

tila	idle	reserving	active	cancelling
syöte				
res_rej	-	idle	cancelling	-
res_ack	-	active	-	-
rel_ack	-	-	cancelling	idle
	reserving	-	cancelling	-
	res_req	res_req	rel_req	rel_req

Palvelimessa ei ole mitään aikavalvontaa eikä käyttöliittymää. Se ainoastaan vastaa asiakkailta tuleviin pyyntöihin.

state	available	allocated	allocated
input		(samalle asiakkaalle)	(eri asiakkaalle)
res_req	allocated	-	-
	res_ack	res_ack	res_rej
rel_req	-	available	-
	rel_ack	rel_ack	rel_ack

Osoitteesta <http://www.tcs.hut.fi/Opinnot/T-79.179/k4/> voi hakea C-kieliset tiedostot `client.c` ja `server.c`, jotka toteuttavat osan edellä kuvatusista algoritmista POSIX-standardin UDP-liitäntää käyttäen. Ne voi kääntää esimerkiksi seuraavilla komennoina:

GNU/Linux:	<code>gcc -o client client.c</code> <code>gcc -o server server.c</code>
Digital UNIX, SGI IRIX:	<code>cc -o client client.c</code> <code>cc -o server server.c</code>
Sun Solaris:	<code>gcc -o client client.c -lsocket</code> <code>gcc -o server server.c -lsocket</code>

Tiedoston `server.c` ohjelma toteuttaa edellä esitetyn palvelimen tilakoneen sellaisenaan. Käynnistä palvelimen tilakone esimerkiksi komennolla `./server 1234`, missä 1234 on jonkin vapaan UDP-portin numero väliltä 1024–65535. Palvelintilakoneen suorituksen voi päättää painamalla **Ctrl-C**.

Tiedoston `client.c` ohjelma käyttää samoja viestejä kuin edellä esitetty asiakkaan tilakone, mutta tilakone on erilainen. Voit kokeilla ohjelmaa komennolla `./client localhost 1234`, missä `localhost` on sen koneen nimi, johon käynnistit palvelintilakoneen ja 1234 on palvelintilakoneen UDP-portin numero. Kokeile välillä pysäyttää palvelintilakone painamalla **Ctrl-Z** ja jatkaa sen suoritusta antamalla komento `fg`. Asiakastilakoneen suoritus päättyy painamalla **Ctrl-D** rivin alussa. Voit käynnistää useita asiakastilakoneita vaikkapa eri tietokoneisiin.

1. Miten tiedoston `client.c` pääsilmutkan (`for (; ;)`) toiminta eroaa edellä esitetystä asiakkaan tilakoneesta? Kuvaa pääsilmutkan toiminta tilakoneena joko graafisesti tai taulukkona.
2. Mihin tarvitaan tiloissa **reserving** ja **cancelling** olevia silmukoita? Millä edellytyksillä silmukat voidaan jättää pois?
3. Muuta tiedoston `client.c` pääsilmutka (`for (; ;)`) vastaamaan annettua asiakkaan tilakoneetta. Ohjelman on ilmoitettava käyttäjälle, milloin se on **idle**-tilassa ja milloin **allocated**-tilassa. Lisäksi sen on otettava vastaan käyttäjän antamia varaus- ja vapautustoimeksiantoja näissä tiloissa. *Lähetä muuttamasi tiedosto `client.c` osoitteeseen marko.makela@hut.fi.*

Palauta tehtävä tietotekniikkatalon huoneiden B 336 ja B 337 väliseen laatikkoon maanantaihin 8.4.2002 klo 16.00 mennessä. Voit myös palauttaa vastauksesi Postscript- tai PDF-muodossa osoitteeseen teemu.tynjala@hut.fi. *Palauta vastauksesi tehtävään 3 osoitteeseen marko.makela@hut.fi.*