Informazio-Sistemen Arkitektura

Telekomunikazio Teknologiaren Ingeniaritzako Gradua. 3. Maila.
Proba praktikoa

2018ko uztailaren 2a Iraupen osoa: 2 ½ h.

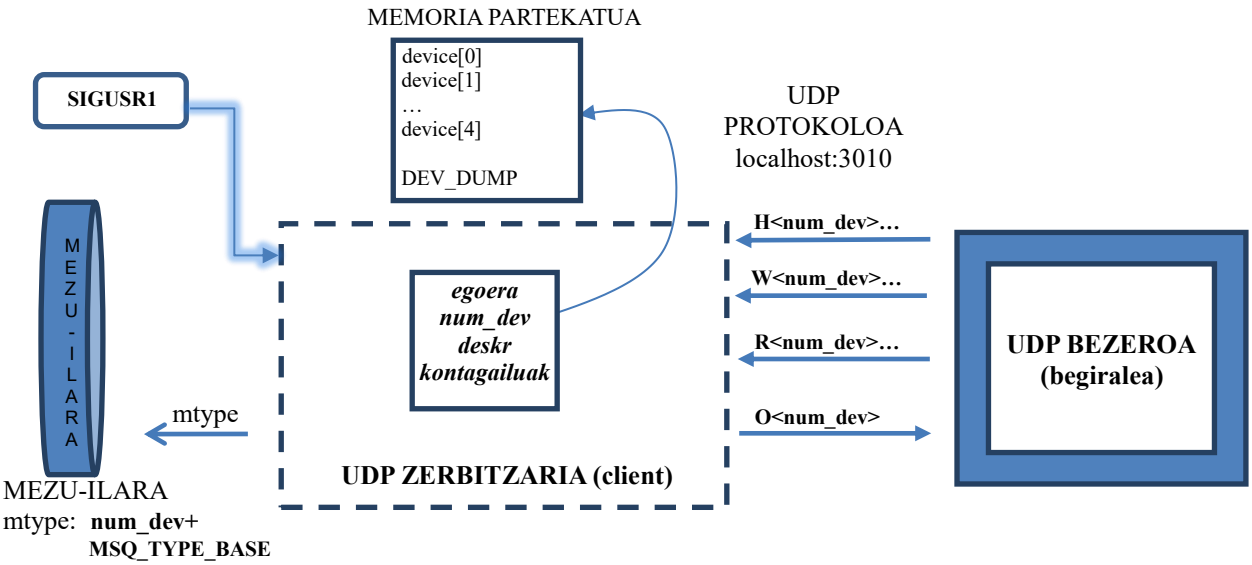
Guztira 15 gako daude. 8 gako jaso behar dira proba gainditzeko

Proba praktikoaren azalpena

Begiraleak UDP bezero bezala lan egingo du eta ikasleak garatuko duen zerbitzarira (*client.c*) konektatuko da zenbait sentsore biomedikori (pultsua, tentsioa, tenperatura, glukosa eta O_2 asetasuna) alta emateko eta berauen neurketak gordetzeko.

Ariketak eta monitorea aurreragoko ariketekin atzeranzko bateragarritasuna izan dezaten diseinatuta daude, hau da, ariketa berriak egiteko ez dago zertan aurreko ariketetan idatzitako kodea borratu ezta komentatu. Honi esker, client aplikazioari funtzionalitate berriak erantsi dakizkioke lanaren garapenean zehar. Atzeranzko bateragarritasun honek ez du esan nahi ariketak euren artean derrigor lotuta daudela. Kontrara, ahal den neurrian modu independentean garatuak izateko pentsatuak izan dira, eta bereziki 5. ariketa modu erabat independentean garatu daiteke gainerako ariketak ia kontuan izan gabe.

Sistemaren arkitektura ondoko irudikoa da:



Memoria partekatuaren lehenengo posizioan hasita sentsore biomedikoen erregistroak gordeko dira, hemendik aurrera gailu bezala adieraziko ditugunak. Gailuaren zenbakiak (*num_dev*) memoria partekatuko gailuen arrayaren indizea adieraziko du. Lehenengo gailua *num_dev*=0 memoria partekatuaren 0 posizioan kokatuko da, honen segidan 1 gailua izango da, eta horrela bost gailu errenkan. Erregistro bakoitzak gailuaren informazioa gordeko du eta baita bere neurriekin lotutako kontagailuen array bat *client_ref.c* fitxategiko *shm_dev_reg* egiturak definitzen duen moduan. Gailu bakoitzari semaforo bana lotuko zaio memoria atzipena sinkronizatzeko.

SIGUSR1 seinalea jasotzean memoria partekatuko gailuen edukia irauli beharko da mezu batez mezu-ilara batera. Memoria partekatuko DEV_DUMP kokapenean aurkituko da irauli beharreko gailu(ar)en indizea. Bere(n) edukia dagokien motako mezuetan kopiatu eta ilarara bialduko dira

SIGUSR1 jasotakoan.

Bi prozesuko hierarkia egitea proposatzen da. Semeak UDP zerbitzaria inplementatuko du eta gurasoak SIGUSR1 seinaleak gobernatuko ditu.

Baliabide guztiek (partekatutako memoriak, semaforoek eta mezu-ilarak) ikaslearen NAN zenbakia erabiliko dute hamaseitarrean eta hizkirik gabe long formatuan (adib: NAN:11234567G, gakoa 0x11234567L) identifikatzeko. Beharrezko baliabideak programaren hasieran irekitzeko gomendioa ematen da.

1. Ariketa. UDP zerbitzariaren inplementazioa

Lehen ariketa honetan oinarrizko UDP komunikazioa ezarri behar da, H (hello) komandoen zain egonez eta O (ok) bidez erantzunez. Garatu behar den aplikazioa localhost:3010 atakan adi egongo da.

H komandoa begiraletik etorriko den datagrama bitar bat izango da, ondoko formatua duena:

H<num_dev>deskr (luzera aldakorra)

Non H byte bat den, *<num_dev>* zenbaki osoak gailuaren zenbakia adierazten duen eta *"deskr"* katea ('\0'z amaitua) gailuaren deskribapena den (< eta > ikurrak ez dira jasotzen). Eskaera honi:

O<num_dev> (5 byte)

erantzunez, **2. gakoa** lortuko da.

Ez bazara gauza H mezuko eremuak ateratzeko eta *O<num_dev>* erantzun zuzena sortzeko, nahikoa izango zaizu socketetik beste edozer mezu erantzutea, adibidez *OK*, **1 gakoa** lortu ahal izateko.

2. Ariketa. Gailuei alta eman memorian

H eskaria jasotakoan memorian dagokion kokapenean erregistratu beharko da *num_dev* gailua, *client-ref.c* fitxategian definituta dagoen *shm_dev_reg* datu-egitura bertan gaurkotuz eta idatziz. *egoera* balioak 1 izan beharko du erregistroa okupatuta dagoela adierazteko, bestela erregistroaren gaurkotzea txarto egin dela ulertuko da. *num-dev*, *deskr* eta *egoera* eremuak egiaztatuko dira.

Lehengo gailuari *num_dev*=0 (0 offseta) alta ondo ematen bazaio, **3. gakoa** lortukoa da. Beste edozein gailuri dagokion kokapenean alta ondo ematen bazaio, **4. gakoa** lortukoa da.

3. Ariketa. Gailuen erregistroak gaurkotu eta irakurri

Ariketa honetan gailuen neurketak memoria partekatuan biltegiratzeko eskariak erantzungo ditugu. Ondoko formatu bitarra dute eskariok:

W<num_dev><kontagailua><balioa> (guztira 13 byte)

non *W* byte bat den, *<num_dev>* zenbaki osoak gailuaren zenbakia adierazten duen, *<kontagailua>* zenbaki osoak arrayaren indizea adierazten duen, eta *<balioa>* zenbaki osoa arrayan gorde beharreko neurria den (< eta > ikurrak ez dira bidaltzen).

W komandotik *num_dev* gailuri dagokion *kontagailua*ren kokapena bilatu beharko da eta bertan idatzi *balioa*. Begiralea *O<num_dev>* erantzunaren zain egongo da. Ondo jasotzen badu erantzun-mezua **5. gakoa** erakutsiko digu.

Begiraleak egingo digun lehen *W* eskaria (*num_dev*=0, *kontagailua*=0) ondo burutuz gero, **6. gakoa** lortuko da. Gainerako *W* eskari-sortari ondo erantzunez gero, **7**. **gakoa**.

Kontagailuen balioak irakurtzeko eskariek ondoko formatu bitarra daukate:

R<num_dev><kontagailua> (guztira 9 byte)

R eskaria jasotakoan *num_dev* gailuaren *kongailua*ren *balioa* irakurri eta ondoko mezu batez erantzun beharko da:

O<num_dev><balioa> (9 byte)

Erantzuna zuzena bada, **8. gakoa** jasoko dugu.

4. Ariketa. Atal kritikoa semaforoen bidez atzitu.

client programak 5 semaforo dituen array bat (begiraleak sorturikoa) atzitu beharko du. Arrayko semaforo bakoitza memoriako gailu banari dagokio hurrenez hurren.

W eragiketak semaforoak zainduz betetzen ba dira **9. gakoa** lortuko da.

R eragiketak semaforoak zainduz betetzen ba dira **10. gakoa** lortuko da.

5. Ariketa. Memoria etenduren eta mezu-ilaren bidez irauli.

Hierarkiako prozesu nagusia izango da SIGUSR1 seinaleak gobernatzeko arduraduna. SIGUSR1 seinale bat jasoko den bakoitzean aplikazioak memoria partekatuko DEV_DUMP kokapenean (bere edukia *num_dev* zenbakia bitarrean izango da) aurkituko duen gailuari dagokion eduki osoa irauli beharko du partekatutako memoriatik mezu-ilarara. *num_dev* aldagaiaren balioak 0 eta 4 artekoak izango dira.

Mezu honen erantzun bezala eskatzen dena zera da: *num_dev* gailuak memorian daukan informazio guztia kopiatu eta mezu-ilarara irauli dezala *num_dev*+MSQ_TYPE_BASE (client_ref.c fitxategian) motako mezu batez. Begiralea gailuaren eduki osoa ondo ekarriko dion *num_dev*+MSQ_TYPE_BASE motako mezu baten zain egongo da.

SIGUSR1-i erantzunez *num_dev*+MSQ_TYPE_BASE motako mezu zuzena jasotakoan **11. gakoa** argitaratuko da.

Gainera, memoria atzitzeko semaforoak ondo erabiltzen badira, **12. gakoa** ere jasoko dugu.

num_dev* = *-1 gailua iraultzeko esaten digutenean, lehenengo bost gailuak hurrenkeran iraultzeko eskatzen zaigu. Kasu honetan bost erantzun-mezu bidali beharko dira, bana gailu bakoitzeko, 0 gailutik hasita 4 gailura artekoak, hurrenkera horretan. Bost erantzun zuzenak jasotzen badira **13. gakoa** erakutsiko zaigu.

***num_dev* = -1** delarik, bost semaforoak batera (0..4) eragiketa bakar batez, eta ez bost eragiketa independenteren bidez, jaisteko gai bagara, **14. gakoa** lortuko dugu.

Amaitzeko, 5. ariketa aurreko ariketetako UDP zerbitzariarekin batera konkurrentzian

exekutatzeko gai bagara, **15. gakoa** lortuko dugu.

Informazio-Sistemen Arkitektura

Telekomunikazio Teknologiaren Ingeniaritzako Gradua. 3. Maila.
Proba praktikoa

2018ko maiatzaren 25a Iraupen osoa: 2 ½ h.

Guztira 15 gako daude. 8 gako jaso behar dira proba gainditzeko

Izena:……
Taldea:……..…… NAN:…….
Gela:……. Ilara:…….. Zutabea:…..... SINADURA:

1 Ariketa

1 Gakoa:	2 Gakoa:

2 Ariketa

3 Gakoa:	4 Gakoa:

3 Ariketa

5 Gakoa:	5 Gakoa:	7 Gakoa:

4 Ariketa

8 Gakoa:	9 Gakoa:

5 Ariketa

10 Gakoa:	11 Gakoa:	12 Gakoa:	13 Gakoa:	14 Gakoa:	15 Gakoa:

client-ref.c izeneko fitxategi bat dago eskuragarri zenbait datu eskaintzen dituena ariketaren espezifikazioetara errazago egokitzeko.