Informazio-Sistemen Arkitektura

Telekomunikazio Teknologiaren Ingeniaritzako Gradua. 3. Maila.
Proba praktikoa

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

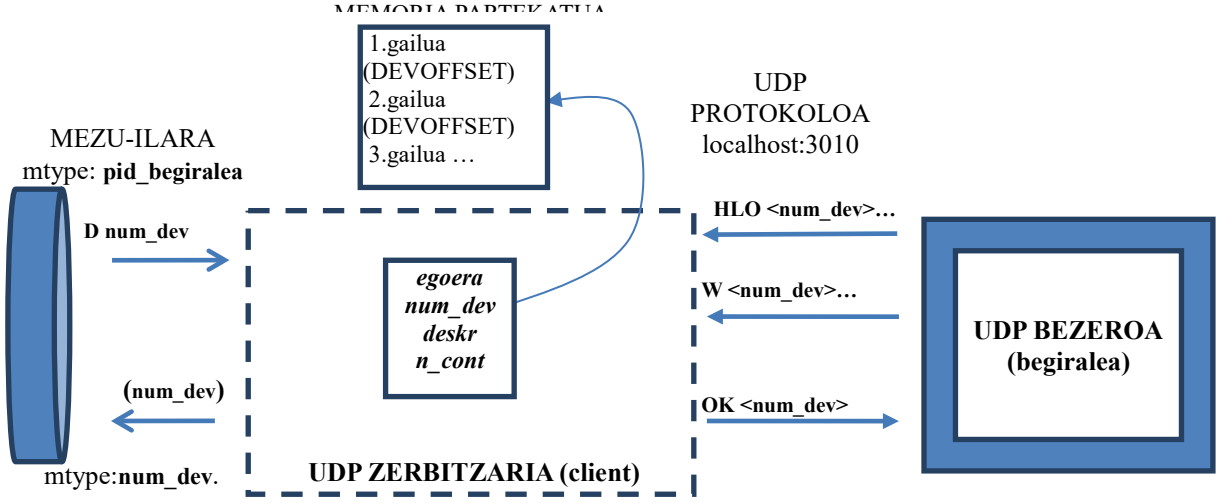
Guztira 15 gako daude. 8 gako jaso behar dira proba gainditzeko

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 eta O_2 asetasuna) alta emateko eta berauen neurketak gordetzeko.

Ariketa hau modu inkrementalean garatuko da, pausu bakoitzean egindakoari funtzionalitate berriak erantsiz eta atzeranzko bateragarritasuna mantenduz. Hau da, 2. ariketa 1. ariketaren hobekuntza izango da, 3.ariketa 2.arena, eta horrela gainerakoekin.

Sistemaren arkitektura ondoko irudikoa da:



Memoria partekatuan gailuen erregistroak gordeko dira, euren artean DEVOFFSET-eko byte-kopuruko saltoa eginez. Lehenengo gailua (*num_dev*=1) memoriaren 0 posizioan kokatuko da, bigarrena (*num_dev*=2) DEVOFFSET-en, hirugarrena DEVOFFSET*2 posizioan eta horrela errenkan. Erregistro bakoitzak gailuaren informazioa gordeko du, baita gailuek bidalitako neurketak ere. *client* programa *HLO* eskarien zain egongo da, gailuei memoria partekatuan alta emateko, eta *W* eskarien zain, sentsoreen neurketak gordetzeko. Jasotako mezu bakoitzari *OK* <*num_dev*> mezu batez erantzungo zaio, non *num_dev* eskarian jasotako gailuaren zenbakia den.

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

1. Ariketa. UDP zerbitzariaren inplementazioa

Lehen ariketa honetan oinarrizko UDP komunikazioa ezarri behar da, *HLO* komandoen zain eta *OK* bidez hauei erantzunez. Garatu behar den aplikazioa localhost:3010 atakan adi egongo da.

HLO komandoa begiraletik testu argian etorriko den datagrama bat izango da, ondoko formatua duena:

HLO <num_dev> <n_cont> deskr

non *num_dev* balioak gailuaren zenbakia adierazten duen ASCII alfanumerikoz, *n_cont* balioak gailuaren kontagailu kopurua ASCII alfanumerikoz eta *"deskr"* katea gailuaren deskribapena den. Jasotako testua ez du bukaeran \0rik izango. Eskaera honi:

OK <num_dev>

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

Ez bazara gauza *HLO* mezuko eremuak ateratzeko eta *OK <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

HLO 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.

Lehengo gailuari *num_dev*=1 (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

Ariketa honetan gailuen neurketak memoria partekatuan biltegiratzeko eskaerak erantzungo ditugu. Eskaera hauek ondoko formatu bitarra dute:

*W <int num_dev> <int **indizea**> <int **balioa**> (13 byte guztira)*

Gailu bakoitzari DEVOFFSET luzerako esparrua dagokio memoria partekatuan, non lehenengo posizioan *shm_dev_reg* datu-egitura gordetzen den eta egitura honen ondoren (int) formatuan sentsorearen neurketak gordetzen diren ondoz ondoko posizioetan. Hortik abiatuta, ***indizea*** eremuak neurketari dagokion posizioa adieraziko digu, eta bertan idatzi beharko dugu neurketaren ***balioa***. Adibidez, baldin *indizea*=0, orduan *balioa num_dev* gailuaren datu-egituraren ondoko lehen neurketan gorde beharko dugu. Baldin *indizea*=1, orduan hurrengo neurketan edo int posizioan idatzi beharko dugu.

W komandotik *num_dev* gailuri dagokion ***indize*** kokapena bilatu beharko da eta bertan idatzi ***balioa*** (int) neurketa. Begiralea *OK <num_dev>* erantzunaren zain egongo da. Ondo jasotzen badu erantzun-mezua **5. gakoa** erakutsiko digu.

Begiraleak egingu digun lehen *W* eskaria (*num_dev*=1, *indizea*=0) ondo burutuz gero, **6**. **gakoa** lortuko da. Gainerako *W* eskari-sortari ondo erantzunez gero, **7**. **gakoa**.

4. Ariketa. Atal kritikoa semaforoen bidez atzitu.

Aplikazioak 5 semaforoko array bat atzituko du (programaren hasieran). Lehengo semaforoa ez da erabiliko eta gainerakoak (1etik 4ra bitartekoak) memoriako gailuetara lotuta egongo dira hurrenez hurren. Adibidez, 4 gailua atzitu nahi izanez gero, 4 semaforoan itxaron beharko dugu.

Semaforoak errespetatuz atzitzen baditugu gailuak, **8. gakoa** lortuko dugu. Kasu honetan, semaforoen egoera leheneratzen badugu **9. gakoa** ere lortuko dugu.

5. Ariketa. Memoria mezu-ilaren bidez irauli.

Ariketa honen funtzionamendua guztiz desberdina da aurreko ariketekin alderatuta. *client* programak aurreko UDP zerbitzariarekin batera, paraleloan beste zerbitzari bat jarriko du berak sortutako mezu-ilaratik etorriko diren eskaerei adi egoteko. Zerbitzari hau begiraletik etorriko diren *pid_begiralea* motako mezuen zain egongo da.

Begiraleak *Dump* mezuak igorriko dizkio gailuek memoria partekatuan duten informazio guztia (erregistroa+neurketak) iraultzeko eskatuz. Hurrengo sintaxia betetzen dute eskariok:

D num_dev

non *num_dev* zenbaki osoa bitarra den. *client-ref.c* erreferentziazko fitxategian definituta dagoen *struct msgq_input* datu-egitura erabili ezazu mezu-ilaratik etorriko diren mezuak irakurtzeko.

Mezu honen erantzun bezala eskatzen dena zera da: *num_dev* gailuak memorian daukan informazio guztia, egitura eta datuak barne (DEVOFFSET arlo osoa), kopiatu eta mezu-ilarara irauli *num_dev* motako mezu batez. Begiralea gailuaren eduki osoa ondo ekarriko dion *num_dev* motako mezu baten zain egongo da. *num_dev* gailu-zenbakiak 1 eta 4 bitartekoak izango dira.

D num_dev eskaria dakarren *pid_begiralea* motako mezu bati, luzera zuzena duen *num_dev* motako mezu batez erantzunez gero, **10. gakoa** bistaratuko du begiraleak.

Erantzun-mezuek dakarten edukia zuzena baldin bada, **11. gakoa** jasoko dugu. Gainera, memoria atzitzeko semaforoak ondo erabiltzen badira, **12. gakoa** ere jasoko dugu.

num_dev = 0 gailua iraultzeko esaten digutenean, lehenengo lau gailuen *Dump-ak* egin ditzagun eskatzen zaigu. Kasu honetan lau erantzun-mezu bidali beharko dira, bana gailu bakoitzeko, 1 gailutik hasita 4 gailura artekoak, hurrenkera horretan. Lau erantzun zuzenak jasotzen badira **13. gakoa** erakutsiko zaigu.

***num_dev* = 0** delarik, lau semaforoak batera (1..4) eragiketa bakar batez, eta ez lau 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.