

Konputagailuen Arkitektura I

Laborategia 1 (ebazpena): etenak

PCaren arkitekturan oinarrituta, dardo-joko bat programatu nahi dugu. Dardo batek ituan (*dianan*) jotzen duenean, gailuak eten bat sortuko du IRQ3 lerrotik, eta datu-erregistroan (0x281 helbidea du) lortutako puntuazioa adieraziko du. Demagun dardo guztiek ituan joko dutela.

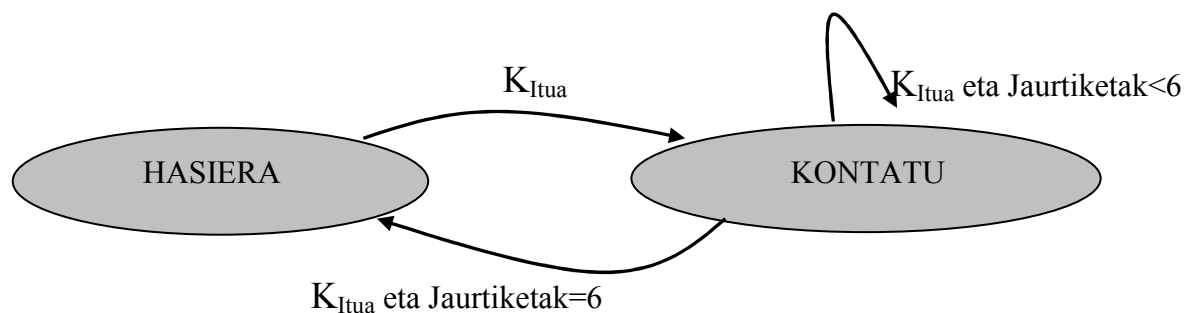
Jokoaren funtzionamendua oso erraza da: 6 jaurtiketatan zein jokalarik lortzen duen puntuazio altuena jakin nahi dugu, erabili duen denbora ere kontuan hartuta. Aurreneko dardoak ituan jotzen duenetik neurtuko dugu denbora. Programaren hasieran, mezu bat aterako da pantailatik, jokalaria joko hasiko dela adierazteko eta dardoak botatzen has dadin. Programa bukatzean, beste mezu bat aterako da pantailatik, lortutako puntuazioa eta behar izan duen denbora (segundoak) adieraziz. Mezuak pantailartzeko, honako funtzio hauek erabiliko ditugu: `IdatzTestu(testu)` eta `IdatzZenbaki(zenb)`.

Azaldutako funtzionamendua kontuan hartuta, idatzi C lengoaiari honako funtzio hauek:

- Programa nagusia. Adierazi programaren funtzionamendurako beharrezkoak diren aldagai globalak.
- Beharrezkotzat jotzen dituzun zerbitzu-errutina guztiak.
- Periferiko baten eskaerak zerbitzatzeko, eten-bektorea aldatzen duen errutina.

Ebazpena

Ariketaren ebazpenerako irudiko automata erabili da. HASIERA egoeran egongo gara lehenengo dardo jaurti eta honek ituan jotzen duen arte. Hortik aurrera, KONTATU egoeran segundoak kontatuko dira eta baita jaurtiketa kopurua ere. Jaurtiketa kopurua 6ra iristen denean berriro HASIERA egoerara itzultzen gara partida bukatu delako.



- Programa nagusia. Adierazi programaren funtzionamendurako beharrezkoak diren aldagai globalak.

Programa nagusian aldagai globalak definitu eta hasieratu behar dira. Baita eten-bektorea aldatu guki idatziko ditugun bi zerbitzu errutinak exekuta daitezten, erlojuaren zerbitzu errutina (eten-bektoreko 0x1C posizioan gorde behar da bere helbidea) eta ituaren zerbitzu errutina (eten bektoreko 0x0B posizioan, 3 eten-lerroari dagokiona, gorde behar da bere helbidea). Horrez gain, programa inoiz bukatzen ez denez, programa nagusian bukatzen ez den begizta bat daukagu. Begizta honetan egiten den gauza bakarra HASIERA egoeran egonez gero (oraindik ez da partida hasi) “hasi jaurtitzen” mezua pantailaratzea da. Bukatzeko eten-bektorearen hasierako egoera berreskuratzeko `BerreskuratuEB` funtzioari deiak dauzkagu. Adibide honetan, hala ere, bi dei horiek ez dira beharrezkoak ez baitira inoiz exekutatu (ikus aurretik inoiz bukatzen ez den begizta bat daukatela).

Automataren egoera gordetzen duen aldagaia (AE) aldagai globala izan behar da, bai programa nagusian eta bai idatzi behar diren bi zerbitzu errutinetan atzitu behar dugulako. Segundoak kontatzeko erabiltzen den aldagaia ere globala izan behar da, erlojuaren zerbitzu errutinan inkrementatu behar delako eta ituaren zerbitzu errutinan pantailaratzen delako jaurtiketa kopurua 6ra iritsi denean. Batura eta jaurtiketa kopurua gordetzen dituzten aldagaiak aldagai globalak izan daitezke edo `static` moduan definitu daitezke ituaren zerbitzu errutinan.

```
int AE;
int Batura, Jaurtiketak, Segundoak;
main()
{
    AE = HASIERA;
    Batura=0;
    Jaurtiketak=0;
    Segundoak=0;
    unsigned short IPoldItua, CSoldItua, IPoldErlojua, CSoldErlojua;

    AldatuEB(0x1C, IP(ZerErruErlojua), CS(ZerErruErlojua),
            &IPoldErlojua, &CSoldErlojua);
    AldatuEB(0x0B, IP(ZerErruItua), CS(ZerErruItua),
            &IPoldItua, &CSoldItua);

    while(1)
        if (AE == HASIERA) IdatzTestu( "hasi jaurtitzen");

    BerreskuratuEB(0x1C, IPoldErlojua, CSoldErlojua);
    BerreskuratuEB(0x0B, IPoldItua, CSoldItua);
}
```

b) Beharrezkotzat jotzen dituzun zerbitzu-errutina guztiak.

Bi zerbitzu errutina idatzi behar ditugu, erlojuaren zerbitzu errutina eta ituaren zerbitzu errutina.

Erlojuaren zerbitzu errutinan KONTATU egoeran, hau da, jokutzen gaudela, pasatzen den denbora kontatu behar da. Horretarako `Tick` aldagaiaren bidez segundo bat noiz pasa den kontatzen dugu (18 eten gertatzen direnean) eta `Segundoak` aldagaia **KONTATU** egoeran segundo bat pasatzen den bakoitzean inkrementatzen dugu.

```
void interrupt ZerErrErlojua()
{
    static int Tick=0;
    Tick++;
    if (Tick==18)
    {
        Tick=0;
        if (AE==KONTATU)
            Segundoak++;
    }
    //IRET;
}
```

Ituaren zerbitzu errutinan kontroladorearen datu-erregistroa irakurri behar da jaurtitako dardoarekin ze puntuazio lortu den jakiteko eta puntuazio hori Batura aldagaian metatu behar da, 6 jaurtiketen guztirako puntuazioa kalkulatzeko. Jaurtiketa kopurua inkrementatu behar da ere. Horretaz gain, hasierako egoeran bagaude, lehenengo jaurtiketa egin dela dakigu, beraz, Segundoak aldagaia hasieratzen dugu kontatzen has dadin eta KONTATU egoerara aldatzen dugu AE aldagaia. Bestela, KONTATU egoeran bagaude eta azken jaurtiketa egin bada (Jaurtiketak==6) informazioa pantailaratu eta aldagai globalak berrasieratzen dira.

```
void interrupt ZerErrItua()
{
    Batura= Batura + InPort(0x281);
    Jaurtiketak ++;
    if (AE==HASIERA)
    {
        AE=KONTATU;
        Segundoak = 0;
    }
    else
    {
        if (Jaurtiketak==6)
        {
            IdatzTestu("Puntuazioa");
            IdatzZenbaki (Batura);
            IdatzTestu("Denbora");
            IdatzZenbaki(Segundoak);
            Batura=0;
            Jaurtiketak=0;
            AE=HASIERA;
        }
    }
    Eoi();
    //IRET;
}
```

c) Periferiko baten eskaerak zerbitzatzeko, eten-bektorea aldatzen duen errutina.

Eten-bektorea aldatzen duen errutinak parametro bezala adierazitako posizioan (sarrera) IPBerria eta CSBerria idatzi behar ditu, aurretik posizio horretako balioak IPZah eta CSZah aldagaietan gordez. Idatzi eta irakurri behar diren balioak 16 bitekoak direnez IrakurHitzFis eta IdatzHitzFis funtzioak erabiltzen dira eten-bektoretik irakurri eta idazteko. Funtzio hauetan lehenengo parametroa eten-bektoreak memorian hartzen duen oinarri helbidea da, 0 segmentua hain zuzen. Atzitzea nahi den eten-bektoreko posizioaren desplazamendua kalkulatzean, berriz, kontuan hartu behar da osagai bakoitzak (IPa gehi CSa) 4 byte okupatzen dituela.

```
void AldatuEB (int sarrera, unsigned short IPBerria,  
              unsigned short CSBerria, unsigned short * IPZah,  
              unsigned short * CSZah)  
{  
    DisableInts(); //Galarazi etenak (IF=0)  
    *IPZah = IrakurHitzFis(0, sarrera*4); //Aurreko IPa gordetzen da  
    *CSZah = IrakurHitzFis(0, (sarrera*4)+2); //Aurreko CSa gordetzen da  
    IdatzHitzFis(0, sarrera*4, IPBerria); // IP berria jartzen da  
    IdatzHitzFis(0, (sarrera*4)+2, CSBerria); // CS berria jartzen da  
    EnableInts(); // Etenak baimendu (IF=1)  
}
```