### **GPSS – LABORATOR 3**

În acest laborator nu se va simula evoluţia unui sistem de aşteptare, ci evoluţia unui sistem care face parte din sistemele de modelare si gestionare a stocurilor. Acestea sunt un alt tip de sisteme pentru care Simularea este o metodă importantă de studiere. Spaţiul de depozitare va fi simulat cu ajutorul unei entităţi STORAGE care de data asta are un rol diferit faţă de cel din laboratorul trecut.

Unele din comenzile şi blocurile noi nu sunt neapărat necesare acestui model, dar sunt folosite în scop didactic. Astfel sunt introduse comenzile EQU şi VARIABLE care definesc constante şi respectiv variabile. Blocul ASSIGN este de asemenea nou şi este folosit pentru a sublinia faptul că tranzacţiile pot avea anumite caracteristici (atribute) ce pot fi memorate şi modificate cu ajutorul acestui bloc în diverşi parametrii ai tranzacţiei.

**Problemă:**

*La un depozit cu o capacitate de 2000 de spaţii de depozitare se fac două operaţii: se aduc materiale în depozit şi se scot materiale din depozit. Iniţial există 1000 de unităţi în depozit.*

* *Depunerea de materiale se face o dată la 5 zile astfel: dacă există mai mult de 800 de unităţi în depozit, depunerea este respinsă, dacă nu, se comanda diferenţa dintre 1000 şi nivelul curent al depozitului. Comanda este onorata in 4 zile.*
* *Scoaterea de materiale din depozit se face în urma unei cereri zilnice de un anumit număr de unităţi, număr care poate lua cu o aceeaşi probabilitate valori întregi între 40 şi 63.*

*Simulaţi evoluţia sistemului pentru 200 de zile. Să se obţină o histogramă care să descrie evoluţia zilnică a numărului de unităţi din depozit.*

**Indicaţii de rezolvare:**

Programul va avea patru părţi:

În prima parte se vor defini anumite entităţi utilizate:

* + Se defineşte depozitul de 2000 de unităţi (vezi laboratorul 2).
  + Se defineşte o histogramă care să afişeze valoarea nivelului zilnic al depozitului (vezi laboratorul 2). De data aceasta operandul A al comenzii TABLE este S$nume\_depozit, B=100, C=100, D=20.
  + Se defineşte o variabilă care să reprezinte numărul de unităţi necesare unei depuneri. Se foloseşte comanda VARIABLE (vezi definiţia de mai jos), S$nume\_depozit pentru nivelul depozitului şi constanta definită mai jos.
  + Se defineşte o variabilă care să reprezinte cererea zilnică. Această variabilă va fi egală cu expresia RN1@24+40. Unde RN1 este generatorul de numere aleatoare cu numărul 1 şi generează numere întregi cuprinse între 0 şi 999. Prin operaţia RN1@24 se generează numere întregi între 0 şi 23.
  + Se defineşte o constantă egală cu numărul de unităţi care reprezintă nivelul iniţial al depozitului. Se foloseşte comanda EQU (vezi definiţia de mai jos).
  + Se defineşte o constantă egală cu numărul minim de unităţi din depozit pentru care o depunere este respinsă. Se foloseşte comanda EQU (vezi definiţia de mai jos).

În a doua parte se va simula procesul de depunere în depozit. După generarea de tranzacţii depunere din 5 în 5 zile se va folosi instrucţiunea TEST cu „variante alternative” (vezi definiţia de mai jos) pentru a verifica dacă nivelul depozitului permite o depunere. Dacă nu, tranzacţia activă va fi trimisă la un bloc TERMINATE. Altfel, folosind instrucţiunea ASSIGN (vezi definiţia de mai jos) i se va atribui parametrului *p1* al tranzacţiei active, cantitatea care se va depune iar apoi urmează simularea depunerii.

În a treia parte se va simula cererea zilnică. Folosind instrucţiunea ASSIGN (vezi definiţia de mai jos) i se va atribui parametrului *p2* al tranzacţiei active, cantitatea care se va scoate din depozit. Se va testa cu instrucţiunea TEST „variante alternative” (vezi definiţia de mai jos) dacă în depozit există un număr suficient de unităţi pentru a îndeplini cererea. Dacă nu, tranzacţia activă „cerere” este distrusă.

În a patra parte se va simula iniţializarea nivelului depozitului. Se va genera o depunere iniţială cu o prioritate maximă. Acesta va consta din 1000 de unităţi care intră în depozit (vezi funcţiile referitoare la STORAGE din laboratorul 2).

**Blocuri şi instrucţiuni folosite:**

**VARIABLE** – Este o comandă care defineşte o entitate reprezentând o variabilă. Forma generală este:

nume VARIABLE X

Unde:

* nume – este o etichetă obligatorie care reprezintă numele variabilei.
* X – este o expresie obligatorie.

**EQU** – Este o comandă care evaluează o expresie. Forma generală este:

nume EQU X

Unde:

* nume – este o etichetă obligatorie reprezentând numele entităţii ce va lua valoarea expresiei X.

**TEST** – Este o instrucţiune bloc care compară valori (de obicei SNA-uri) şi controlează destinaţia tranzacţiei active pe baza rezultatului comparaţiei.

Forma generală este:

## **TEST O A,B,C**

Unde:

* O – este un operator relaţional obligatoriu . Pentru un test cu succes, trebuie să se verifice relaţia A O B. Operatorul O poate să ia valorile: E, G, GE, L, LE, sau NE. Dacă este:
* E – pentru un test cu succes, trebuie ca A=B.
* G - pentru un test cu succes, trebuie ca A>B.
* GE - pentru un test cu succes, trebuie ca AB.
* L - pentru un test cu succes, trebuie ca AB.
* LE - pentru un test cu succes, trebuie ca AB.
* NE - pentru un test cu succes, trebuie ca AB.
* A – Reprezintă valoarea testată şi este obligatoriu. Poate să fie nume, număr, string, expresie cu paranteze, SNA sau parametru SNA.
* B – Reprezintă valoarea de referinţă şi este obligatoriu. Poate să fie nume, număr, string, expresie cu paranteze, SNA sau parametru SNA.
* C – Reprezintă eticheta blocului de destinaţie şi este opţional.

Un bloc TEST lucrează fie în modul “refuz” fie în modul “variante alternative”. În ambele cazuri sunt evaluaţi şi comparaţi operanzii A şi B.

Dacă operandul C nu este folosit, blocul TEST lucrează în modul “refuz”. Când o tranzacţie încercă să intre într-un astfel de bloc, şi testul nu are succes, atunci tranzacţia este blocată, adică nu i se permite să intre în blocul TEST până când testul se repetă şi are succes. Când testul are succes, tranzacţia activă intră în blocul TEST parcurgând apoi secvenţial blocurile modelului.

Dacă operandul C este folosit, blocul TEST lucrează în modul “variante alternative”. Când o tranzacţie încearcă să intre într-un astfel de bloc TEST, iar testul este fără succes, ea este trimisă la blocul cu eticheta C. Dacă testul are succes, tranzacţia parcurge în mod secvenţial blocurile din model care urmează blocului TEST.

**ASSIGN** – Este o instrucţiune bloc folosită pentru a atribui sau a modifica valoarea unui parametru al tranzacţiei. Forma generală este:

## **ASSIGN A,B,C**

Unde:

* A – este numărul parametrului tranzacţiei active. Este obligatoriu şi poate fi un nume, un întreg pozitiv, o expresie cu paranteze, SNA, sau parametru SNA, câteodată fiind urmat de + sau -.
* B – este o valoare obligatorie care poate fi nume, număr, string, expresie cu paranteze, SNA sau parametru SNA.
* C – este un număr de funcţie şi este opţional. Poate fi nume, întreg pozitiv, expresie cu paranteze, SNA sau parametru SNA.

Exemple:

## **ASSIGN 2000,150.6**

Valoarea 150.6 este atribuită parametrului tranzacţiei active, care are numărul 2000. Dacă nu există un astfel de parametru, atunci este creat.

## **ASSIGN TEXT,”kmlm.,m.,lafdsaf”**

Parametrului tranzacţiei active cu numele TEXT îi este atribuit stringul scris între ghilimele. Dacă nu există un astfel de parametru, el este creat.

## **ASSIGN 2000+,3**

În acest exemplu, semnul + care urmează operandul A indică faptul că valoarea operandului B se adaugă valorii parametrului 2000 al tranzacţiei active. Dacă un astfel de parametru nu există, atunci este creat şi iniţializat cu 0 înainte de adunare.

## **ASSIGN 2000-,3**

În acest exemplu, semnul - care urmează operandul A indică faptul că valoarea operandului B se scade din valoarea parametrului 2000 al tranzacţiei active. Dacă un astfel de parametru nu există, atunci este creat şi iniţializat cu 0 înainte de scădere.

**Rularea modelului de simulare:**

* Rulaţi programul pentru 200 de zile.
* Urmăriţi în fereastra expresiilor numărul de cereri prea mari. La câmpul “Expression” treceţi: N$Etichetă\_bloc. Unde Eichetă\_bloc este eticheta blocului unde se vor distruge tranzacţiile de tip cerere care sunt prea mari.
* Procedaţi analog pentru numărul maxim de unităţi care sunt în depozit. La “Expression” se scrie SM$nume\_depozit.
* Procedaţi analog pentru numărul de unităţi din depozit. La “Expression” se scrie S$nume\_depozit.

Pentru un model descris de un program în GPSS există două SNA-uri care se referă la timpul simulării: C1 si AC1. C1 este timpul relativ al simulării, iar AC1 este timpul absolut al simulării. Să se urmărească în fereastra expresiilor modificarea celor două valori în timpul simulării. Ce observaţi? Să se aplice comanda RESET şi apoi să se observe din nou valorile timpilor C1 şi AC1. Ce observaţi? Să se aplice comanda CLEAR şi să se repete observaţia de mai sus. Să se caute în HELP definiţia comenzilor RESET şi CLEAR.

**Problemă suplimentară:** modificaţi programul astfel încât să existe cereri care să depăşească numărul unităţilor existente la un moment dat în depozit.