

GPSS – LABORATOR 3

În acest laborator nu se va simula evoluția unui sistem de așteptare, ci evoluția unui sistem care face parte dintre sistemele de modelare și gestionare a stocurilor. Acestea sunt un alt tip de sisteme pentru care Simularea este o metodă importantă. Spațiul de depozitare va fi simulat cu ajutorul unei entități STORAGE care de data aceasta are un rol diferit față de cel din laboratorul trecut.

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

Problemă:

La un depozit cu o capacitate de 2000 de spații de depozitare se fac două operații: se aduc materiale în depozit și se scot materiale din depozit. Inițial există 1000 de unități în depozit.

- Depunerea de materiale se face o dată la 5 zile astfel: dacă există mai mult de 800 de unități în depozit, depunerea este respinsă, dacă nu, cantitatea depusă reprezintă diferența dintre 1000 și nivelul depozitului. Depunerea se realizează după 5 zile.*
- Scoaterea de materiale din depozit se face în urma unei cereri zilnice de un anumit număr de unități, număr care poate lua cu o aceeași probabilitate valori întregi între 40 și 63.*

Simulați evoluția sistemului pentru 200 de zile. Să se obțină o histogramă care să descrie evoluția zilnică a numărului de unități din depozit (repartiția numărului de unități din depozit determinat la sfârșitul fiecărei zile de lucru).

Studiați următoarele două cazuri:

- depunerea are o prioritate mai mare decât cererea*
- cererea prezintă o prioritate mai mare decât depunerea.*

Indicații de rezolvare:

Programul va avea patru părți:

- În prima parte se vor defini anumite entități utilizate:
 - Se definește depozitul de 2000 de unități (vezi laboratorul 2).
 - Se definește o histogramă care să afișeze distribuția nivelului zilnic al depozitului (vezi laboratorul 2). De data aceasta operandul A al comenzii TABLE este

S\$*nume_depozit* (atribut care returnează numărul unităților din entitatea storage cu eticheta “*nume_depozit*” ocupate), B=100, C=100, D=20.

- Se definește o variabilă care să reprezinte numărul de unități necesare unei depuneri. Se folosește comanda VARIABLE (vezi definiția de mai jos), S\$*nume_depozit* pentru nivelul depozitului și constanta definită mai jos.
 - Se definește 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 și generează numere întregi cuprinse între 0 și 999. Prin operația [RN1@24](#) se generează numere întregi între 0 și 23.
 - Se definește o constantă egală cu numărul de unități care reprezintă nivelul inițial al depozitului. Se folosește comanda EQU (vezi definiția de mai jos).
 - Se definește o constantă egală cu numărul minim de unități din depozit, pentru care o depunere este respinsă. Se folosește comanda EQU (vezi definiția de mai jos).
2. În a doua parte se va simula procesul de depunere în depozit. După generarea de tranzacții depunere din 5 în 5 zile se va folosi instrucțiunea TEST cu „variantă alternativă” (vezi definiția de mai jos) pentru a verifica dacă nivelul depozitului permite o depunere. Dacă nu, tranzacția activă va fi trimisă la un bloc TERMINATE. Altfel, folosind instrucțiunea ASSIGN (vezi definiția de mai jos) i se va atribui parametrului *p1* al tranzacției active, cantitatea care se va depune iar apoi urmează simularea depunerii.
 3. În a treia parte se va simula cererea zilnică. Folosind instrucțiunea ASSIGN (vezi definiția de mai jos) i se va atribui parametrului *p2* al tranzacției active, cantitatea care se va scoate din depozit. Se va testa cu instrucțiunea TEST „variantă alternativă” (vezi definiția de mai jos) dacă în depozit există un număr suficient de unități pentru a îndeplini cererea. Dacă nu, tranzacția activă „cerere” este distrusă.
 4. În a patra parte se va simula inițializarea nivelului depozitului. Se va genera o depunere inițială cu o prioritate maximă. Acesta va consta din 1000 de unități care intră în depozit (vezi funcțiile referitoare la STORAGE din laboratorul 2).

Blocuri și instrucțiuni folosite:

VARIABLE – Este o comandă care definește 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.

Valoarea expresiei X va fi calculată și atribuită variabilei definite prin eticheta *nume*, cu comanda VARIABLE, folosind atributul V\$ *nume*.

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

nume EQU X

Unde:

- nume – este o etichetă obligatorie reprezentând numele entității ce va lua valoarea expresiei X.

TEST – Este o instrucțiune bloc care compară valori (de obicei SNA-uri) și controlează destinația tranzacției active pe baza rezultatului comparației.

Forma generală este:

TEST O A,B,C

Unde:

- O – este un operator relațional obligatoriu . Pentru un test cu succes, trebuie să se verifice relația 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 $A\geq B$.
 - L - pentru un test cu succes, trebuie ca $A<B$.
 - LE - pentru un test cu succes, trebuie ca $A\leq B$.
 - NE - pentru un test cu succes, trebuie ca $A\neq B$.
- A – Reprezintă valoarea testată și este obligatoriu. Poate să fie nume, număr, string, expresie cu paranteze, SNA sau parametru SNA.

- B – Reprezintă valoarea de referință și este obligatoriu. Poate să fie nume, număr, string, expresie cu paranteze, SNA sau parametru SNA.
- C – Reprezintă eticheta blocului de destinație și este opțional.

Un bloc TEST lucrează fie în modul “refuz” fie în modul “variante alternative”. În ambele cazuri sunt evaluați și comparați operanzii A și B.

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

Dacă operandul C este folosit, blocul TEST lucrează în modul “variante alternative”. Când o tranzacție 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, tranzacția parcurge în mod secvențial blocurile din model care urmează blocului TEST.

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

ASSIGN A,B,C

Unde:

- A – este numărul parametrului tranzacției active. Este obligatoriu și 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 funcție și este opțional. Poate fi nume, întreg pozitiv, expresie cu paranteze, SNA sau parametru SNA.

Valoarea parametrului asociat tranzacției poate fi aflată folosind atributul “pA”, dacă A este un număr sau “p\$A”, dacă A este un string.

Exemple:

ASSIGN 2000,150.6

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

ASSIGN TEXT,"kmlm.,m.,lafdsaf"

Parametrului tranzacției 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 tranzacției active. Dacă un astfel de parametru nu există, atunci este creat și inițializat 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 tranzacției active. Dacă un astfel de parametru nu există, atunci este creat și inițializat cu 0 înainte de scădere.

Rularea modelului de simulare:

- Rulați programul pentru 200 de zile.
- Urmăriți în fereastra expresiilor numărul de cereri prea mari. La câmpul "Expression" treceți: N\$Etichetă_bloc. Unde Etichetă_bloc este eticheta blocului unde se vor distruge tranzacțiile de tip cerere care sunt prea mari.
- Procedați analog pentru numărul maxim de unități care sunt în depozit. La "Expression" se scrie SM\$nume_depozit.
- Procedați analog pentru numărul de unități 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 și 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 observați? Să se aplice comanda RESET și apoi să se observe din nou valorile timpilor C1 și AC1. Ce observați? Să se aplice comanda CLEAR și să se repete observația de mai sus. Să se caute în HELP definiția comenzilor RESET și CLEAR.

Problemă suplimentară: modificați programul astfel încât să existe cereri care să depășească numărul unităților existente la un moment dat în depozit.