Bul.St.Univ.Baia Mare
Seria B, Matematică-Informatică, Vol.VII, Nr.1-2, 153-159

PROCESOARE DE MESAJE

Radu SIMA

1. Prezentare generala.

Programarea "obiectuala" este o necesitate in contextul diversificarii si profesionalizarii aplicatiilor de sistemele de calcul. Din acest punct de vedere, exista tendinte de implementare a unor aplicatii in care traficul de mesaje sa nu mai fie un simplu vad pentru lucrarea cu texte, ci un complex integrat de functii, in general multi-media. Un astfel de exemplu il constituie implementarile "hipertext" pe calculatoare APPLE.

Cu toate acestea, inca nu s-a raportat un mediu de programare pentru astfel de implementari, cele mai sofisticate realizari in acest domeniu fiind proiecte dedicate. In aplicatiile curente, mesajele sunt simple argumente restructurate ale unor functii de limbaj.

Studiul de fata il constituie un astfel de mediu integrator al mesajelor, vazute ca obiecte structurate independente de context.

2. Conditii preliminare.

Restrictiile care se pun unei probleme de tip flux de mesaje sint urmatoarele:

- un numar nelimitat de mesaje in sistem;
- trafic foarte intens de mesaje intre sistem si operator;
- necesitatea unei viteze foarte mari de raspuns;
- imposibilitatea predictarii cererii unui mesaj;
- necesitatea prezentarii in diferite videoformate, grafica, sonorizare, etc. a informatiei;
- recunoasterea codurilor pentru intrega clasa de programe ce sint deservite de acelasi procesor de mesaje;
- posibilitatea modificarii rapide a mesajelor, actualizarea lor independent de programele de aplicatie care le utilizeaza;
- independenta totala codului programelor fata de mesajele pe care le solicita;
- portabilitatea si translabilitatea in orice limba.

Aceste restrictii impun de la inceput realizarea unei resurse speciale a sistemului care sa gestioneze in siguranta traficul de mesaje, intr-un mod general si integrat. Aceste trebuie sa opereze cu structuri bine definite, livrind programelor (tasklor) solicitante fie o forma "despachetata" a obiectului, fie un cod reprezentind modul in care sarcina a fost indeplinita (in cazul in care programul apelant solicita executia unei functii).

3. Principiile de baza ale procesorului de mesaje MP (MESSAGE PROCESSOR)

Din cele aratate mai sus rezulta in primul rind necesitatea separarii nete a programului de aplicatie fata de codul procesorului de mesaje si mesajele propriuzise.

Primul deziderat poate fi realizat prin mai multe metode, in functie de natura aplicatiei care se construieste. Cea mai simpla este inglobarea intregului MP cu codul executabil. Acest mod este practic numai la aplicatiile mici care vor sa profite de tezaurului gata fabricat al unor aplicatii mari, precum si de functiile integrate ale MP. Procedeul este simplu, existe module de legatura intre diferitele limbaje uzuale gazda si MP, acestea din urma fiind scrise in "C".

Pentru aplicatiile mari, fie multitasking, fie cu multe programe "independente", sau in mediile integrate asemanatoare suprafetelor grafice, modul prezentat mai sus este impropriu: diseminarea unor module identice in fiecare aplicatie ar fi o risipa inutila pe de o parte, iar pe de alta parte, nu ar putea fi folosite capacitatile foarte rafinate de cautare si livrare a solicitarilor. In acest caz, MP se instaleaza rezident de tip TSR (TERMINATE AND STAY RESIDENT) putind fi cuplat prin apeluri (de tip intreruperi sau "filtre") depinzind de sistemul de operare pe care este prefat. Consecintele sint imediate:

- programele de aplicatie sint alcatuite numai din cod pur;
- oricare program poate avea acces la orice mesaj din sistem, acestea fiind un bun comun al ansamblului;
- MP poate fi interogat contextual, de exemplu prin actionarea unor "teste calde", mod deosebit de util la fabricarea "help-urilor instant" pe calculatoarele PC.

Mesajele sint inglobate intr-un fisier unic cu structura speciala, care este accesat de MP. Prin acest mod se realizeaza dezideratul de universalitate al mesajelor relativ la apelant. O foarte importanta consecinta rezida de aici: intr-o aplicatie, mesajele pot fi generate asincron fata de generarea codului. Astfel, ele pot lua o forma definitiva doar dupa ce intreaga aplicatie a fost realizata si testata profund.

Fiecare mesaj este nominalizat printr-un cod. Acesta se rezuma in ultima instanta la numarul de ordine in fisierul de mesaje, ceea ce nu impiedica existenta unor tabele de translatare intre numarul de ordine si un criteriu literal extern procesorului.

Livrarea unui mesaj presupune regasirea lui, formatarea sau structurarea si livrarea lui fie spre apelant fie spre poarta de iesire (videoterminalul, de exemplu). Apelantului i se remite un cod de eroare in cazul in care cererea nu a putut fi indeplinita. Deoarece apelantul nu are cunostiinta aprioricа despre continutul mesajului solicitat, rezulta ca acesta poate contine orice: de la texte ordinare pina la structuri complexe de programare, coduri executabile, imagini grafice, overlay-uri, etc. Astfel, este realizat si conceptul de mesaj - obiect, operatiile solicitate reprezentind cerinte principiale, operatiile executate fiind in functie de content.

O cerinta drastica pe care trebuie sa o indeplineasca MP este livrarea operativa a mesajelor. In unele aplicatii aceasta restrictie este foarte drastica: in sistemele multi user spre exemplu, sau in aplicatiile CAM sau supervizor in ceea ce priveste raspunsurile la conditii de avarie.

Mecanismul de acces la un mesaj este strins legat de structura fisierului de mesaje precum si de conditiile de instalare a procesorului MP. In fisier, mesajele sint grupate in arii de mesaje, ca unitate minima de acces la fisier. Ariile pot avea dimensiuni foarte variate, depinzind de taskurile care le folosesc, etc. In general, intr-o astfel de arie sint inglobate mesaje cu aceeasi etnie, de exemplu apartinatoare aceleiasi clase de taskuri. Memoria de lucru, TMAGEN (TMA GENERAL) este divizata in partitii de dimensiune variabila, numite TMA (TRANSITORY MESSAGE AREA). Dimensiunea minima a unei TMA nu poate fi mai mica decit dimensiunea ariei minime din fisierul de mesaje. Ariile de mesaje care contin mesajele solicitate se vor incarca succesiv in TMA-uri, depinzind de modul de operare al MP. Independent de mod, algoritmul de livrare al unui mesaj este urmatorul: se cauta mesajul in TMA-urile ocupate. Daca nu este gasit, se incarca aria mesajului solicitat in una din TMA-uri, eliberind-o pe cea mai putin prioritara. Prioritatea este dependenta de modul in care opereaza MP.

Din acest punct de vedere, MP deosebeste mai multe categorii de aplicatii, in care livrarea mesajelor se optimizeaza in functie de privilegiu mesajului, de probabilitatea de solicitare a mesajului sau de privilegiul taskului apelant. In acest sens se pot delimita urmatoarele configuratii de baza:

- aplicatiile cu mesaje ale caror aparitii sint echiprobabile cu taskuri uniform privilegiate: modul UNIFORM;
- aplicatii cu mesaje echiprobabile dar cu taskuri ierarhizate dupa un criteriu de prioritate: modul PRIORITY;
- aplicatii cu mesaje privilegiate, indiferent de probabilitatea de aparitie sau de prioritatea taskurilor apelante: modul PRIVILEGE;
- aplicatii cu mesaje distribuite dupa o statistica de probabilitate oarecare, cu taskuri uniform prioritare: modul PROBABILITY.

Evident, MP poate opera si cu combinatii de astfel de situatii standard, in orice forma imaginabila.

In modul UNIFORM, fiecarui task i se aloca una sau mai multe zone de memorie TMA dedicate, in bufferul TMAGEN. Solicitarea unui mesaj presupune cautarea acestuia in oricare din ariile deja existente in TMA, iar daca acesta nu este prezent, incarcarea ariei apartinatoare in zona TMA alocata taskului. Cind sint mai multe taskuri decit zone in TMA, acestora li se vor atribui spre folosinta zone comune, taskurile fiind grupate in clase de echivalenta relativ la mesaje: de exemplu, acele care folosesc in comun mesaje sau care sint indiferente relativ la durata accesului la un mesaj. Sistemul de reincarcare a ariilor corespunzatoare taskurilor din clase de echivalenta similare este de tipul "zona dedicata". Taskuri din clase de echivalenta diferite nu se vor "stinjeni" intre ele relativ la refolosirea memoriei. Daca toate taskurile sint in aceeasi clasa de echivalenta, metoda de alocare folosita este rotativa, ROUND ROBIN.

In modul PRIORITY, tehnica de accesare a unui mesaj este similara cazului descris anterior, cu observatia ca eliberarea unei zone pentru aducerea ariei de pe suport in memorie se face dupa un criteriu de prioritate: se va elibera aria corespunzatoare clasei taskurilor celor mai putin prioritare. In acest sens, prioritatea se stabileste ca un criteriu de instalare al aplicatiei, pe baza analizei procesului pe care il implementeaza. In acest mod, taskurile foarte prioritare au ariile de mesaje gata pregatite in memorie, cu maximum de probabilitate.

In modul PRIVILEGE, exista mesaje prioritare, care se cer accesate instantaneu. In aceasta situatie, MP deschide o TMA speciala, in care sint recoltate la lansarea aplicatiei toate mesajele specificate intr-o lista ca parametru de apel. Acest mod garanteaza existenta in memorie a acestor mesaje, indiferent de ariile din fisier unde isi au obirsia. Aceste mesaje pot contine informatii de avarie, cuvinte de dictionar foarte des utilizate, structuri de tip meniu, etc.

In modul PROBABILITY, nu exista apriori nici un criteriu de preferentiere a taskurilor sau a mesajelor; un analizor statistic calculeaza in mod experimental probabilitatea de solicitare al unui mesaj. Se elibereaza din memorie acele TMA-uri care au cea mai mica probabilitate de solicitare al mesajelor care o contin. Se observa ca acest mecanism este similar unei memorii CACHE si deci functioneaza ca atare. Este modul implicit, folosit in special atolo unde nu se cer criterii de prioritate ale taskurilor sau mesajelor.

In practica, se pot combina aceste moduri. Cea mai des intrebuintata metoda este imbinarea modului PRIVILEGE cu PROBABILITY, care da rezultate foarte satisfacatoare in majoritatea aplicatiilor mari si foarte mari.

De remarcat ca notiunea de task este mai larga aici, un singur program poate accesa procesorul de mesaje, anumite resurse ale lui putind fi asimilate unor taskuri separate. Se poate

realiza astfel o dihotomie intre mesajele de help de exemplu si mesajele curente dintr-o faza anumita a executiei programului.

Din punctul de vedere al optimizarii spatiului de memorie, MP poseda un sistem dinamic de alocare, cu un sistem autentic GARBAGE COLLECTOR, pentru eliberarea interstitiilor neocupate. In acest mod, TMA-urile pot avea o dimensiune reala mai mare decit cea fixata prin parametru de apel, ceea ce conduce la o mai buna optimizare de acces la un mesaj.

4. Structura fisierului de mesaje. Structura mesajelor.

Deoarece numarul mesajelor este nelimitat, iar dimensiunea unui mesaj este imprevizibila, s-a ales o structura complexa a fisierului de mesaje. Fara a intra in detalii, se poate rezuma faptul ca acesta contine un director de lungime fixa, fiecare intrare fiind alocata unei arii. In aceasta intrare se gasesc informatii relative la mesajele pe care le contine aria si dimensiunea ei. O parcurgere secventiala a directorului conduce la determinarea offset-ului ariei fata de inceputul fisierului si la numarul de octeti care trebuie incarcati in memorie. Directorul poate fi pastrat permanent in memorie intr-o zona DIR a MP-ului.

In interiorul ariilor, mesajele sint secventiale, avind ca informatie de regasire dimensiunea fiecaruia. In acest fel, orice mesaj este accesat printr-un numar similar de operatii (deci indiferent de dimensiunea si numarul de mesaje din aplicatie.

In corpul mesajelor se gasesc texte propriu-zise precum si cuvinte de control necesare functiilor pe care le realizeaza MP. Textele sint livrate fara nici o modificare din partea procesorului. Cuvintele de control definesc urmatoarele grupe mari de functii:

- inserarea unui sir de substituire: MP substituie cuvintul de control cu sirul a carei adresa este specificata ca parametru de apel;
- inserare mesaj: se substituie cuvintul de control cu mesajul a carui numar este specificat ca parametru de apel. In acest sens, MP functioneaza recursiv, limitarile fiind date doar de dimensionarea stivei interne. Prin aceasta functie, se pot construi texte oricit de lungi, plecind de la un "dictionar" cu citeva mii de cuvinte, ca mesaje elementare;
- functii de terminal: se modifica fontul, culorile, dimensiunile textului ce urmeaza. Evident, aceasta operatiune se face in conjunctie cu o alta resursa necesara in aplicatii si anume cu un procesor de ecran (SCREEN MANAGER);
- creare fereastra: se livreaza parametri pentru generarea unei ferestre fizice (pe PC) sau logice (pe mini);
- generare meniuri (bara, verticale): se livreaza apelantului intraga structura a meniului, precum si procedura de

selectie. Se returneaza pointerul catre functia selectata;
- generare de machete si validare pentru introducerea datelor: se specifica pozitiile, textele insotitoare si criteriile de validare ale datelor introduse prin machete;
- atasarea help-ului automat (pe mai multe nivele) unei secvente de dialog operator;
- transmitere octeti nedefiniti: pe o dimensiune de lungime specificata, apelantul receptioneaza un sir de octeti pe care MP nu il interpreteaza. In acest mod se pot transmite segmente de cod, imagini, etc.

Combinind toate cuvintele de control, se pot realiza structuri cu un grad inalt de complexitate, astfel ca de multe ori aplicatia propriu-zisa rezida dintr-un numar de apeluri catre MP, acesta preluind intreaga sarcina: de la ferestre si meniuri, la help automat, grafica, etc. Tinind cont de diferitele modalitati de multitasking, se pot concepe aplicatii cu orice criteriu de optim imaginat.

5. Servicii speciale. Limbajul de generare (MG).

Pentru partea de dezvoltare a fost creat un limbaj de constructie a mesajelor, numit MG (MESSAGE GENERATOR). Cuvintele cheie (30) se leaga intr-o sintaxa "C", sursa fiind compilata, linkeditata si executata. In urma executiei se creaza / modifica fisierul de mesaje. Avantajele sint imediate: usurinta cu care se pot crea structuri oricit de complicate, obiecte numite generic "mesaje". Sintaxa este simpla, nu se cere nici un fel de cunostinte apriorice asupra limbajului gazda. Principalul dezavantaj rezida in faptul rezultatele creatiei nu se vad decit dupa generarea fisierului, deci nu se incadreaza in grupa aplicatiilor WYSIWYG, asa cum sint editoarele de texte evoluate sau aplicatiile DTP. Acest fapt poate provoca unele neajunsuri prin corectii OFF LINE si repetari ale generarii atunci cind machetele gindite nu corespund cu ceea ce se doreste in final.

In faza de punere la punct, clauza TEST provoaca inglobarea unui depanator care ajuta mult la stabilirea corectiilor datorate unor eventuale greseli de apel. O bogata gama de mesaje de eroare si avertizare insotesc executia MP sub depanator.

Procesorul de mesaje descris mai sus a fost folosit cu succes la implementarea mesajelor pentru programul de informatizare globala al Bibliotecii Nationale Romane.

B I B I O G R A F I E

"FOLID" - Programe pentru Biblioteca Nationala. Rezumate la Simpozionul de informatica in documentare, Predeal, noiembrie 1988. I.C.S.T.C.I.

UNIVERSITY OF BAIA MARE
4800 Baia Mare
ROMANIA