

# O IMPLEMENTARE MULTITASKING A STANDARDULUI GKS

Ec. Costin Pribeanu  
Mat. Miti Lucian Ionescu

Institutul de Cercetări în Informatică

## Rezumat

Articolul prezintă o serie de considerații privind o implementare de tip multitasking a standardului grafic GKS pe minicalcatoare, semnificând totodată impactul pozitiv al acestei abordări asupra portabilității sistemelor grafice.

## 1. Introducere

Standardul GKS cuprinde un set larg de funcții destinate dezvoltării la un nivel ridicat de eficiență și portabilitate a aplicațiilor grafice.

Dificultățile ridicate de implementarea sa, datorate complexității și numărului mare de rutine care trebuiau să acopere funcționalitatea, au impus o analiză prealabilă atentă a arhitecturii viitorului sistem.

Următoarele cerințe, derivate din particularitățile mediului hard-soft și din specificul aplicațiilor grafice, au determinat structura pachetului grafic:

- sistemul de calcul pe care urma să ruleze este un minicalculator din familia PDP 11 (CORAL, INDEPENDENT) funcționând sub sistem de operare RSX;
- mărimea task-urilor care pot rula pe aceste sisteme este limitată la 64 ko; furnizarea funcțiilor GKS sub forma unei biblioteci de rutine ar greva programul de aplicație de un spațiu de memorie considerabil;
- atingerea funcționalității cerute de nivelul 2b presupune implementarea conceptului "multiple stații de lucru";
- obținerea unui grad înalt de portabilitate printr-o interfațare convenabilă între partea independentă de dispozitiv - nucleul grafic și partea dependentă de stațiile de lucru - circuitul de comandă (device-driver);
- realizarea cu ușurință a interfeței cu programul de aplicație care poate fi scris și în alt limbaj decât pachetul de rutine grafice.

## 2. Structura software

Dezideratul de a asigura o independență pe două niveluri a programului de aplicație față de nucleul de rutine independente de dispozitiv și a nucleului față de circuitele de comandă ale stațiilor de lucru a condus la

o implementare de tip multitasking.

Sistemul se oferă programatorului de aplicație în următoarea structură:

- un set de biblioteci de interfață, rutinele fiind înglobate în task-ul de aplicație;
- un task nucleu;
- un set de task-uri pentru fiecare din echipamente sau clasele de echipamente grafice disponibile.

În timpul rulării unei aplicații grafice, configurația cuprinde task-ul de aplicație, task-ul nucleu și câte un task pentru fiecare stație de lucru deschisă (v. fig. 1).

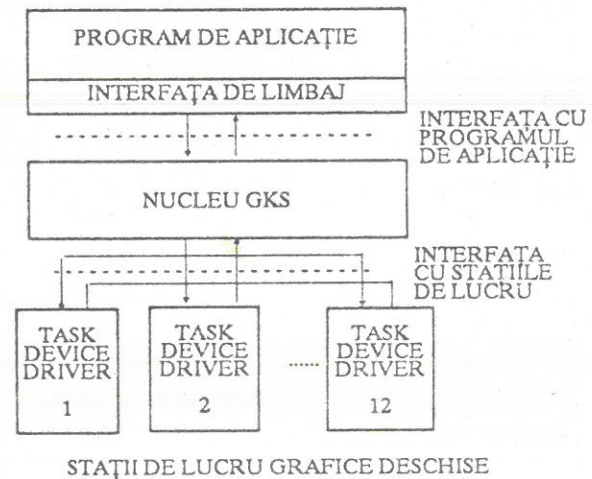


Figura 1

Sincronizarea task-urilor se face prin flag-uri de grup, comunicarea fiind asigurată de o zonă de date de tip COMMON RESIDENT.

Zona de date cuprinde informații necesare celor trei task-uri: lista GKS, o parte din lista segmentelor și un set de buffere care asigură o cale de comunicație bidirecțională între cele trei niveluri ale implementării.

## 3. Interfața utilizator

Una din problemele pe care trebuie să le rezolve o implementare de standard GKS este proiectarea interfeței pachetului grafic cu programul de aplicație. Felul în care este abordată această sarcină, precum și efortul pe care îl implică sînt determinate de modalitatea în care nucleul este legat de aplicația grafică.

Furnizarea nucleului sub forma unei biblioteci de rutine necesită numai implementarea limbajului de legătură, dar prezintă dezavantajul unei dependențe a programului de aplicație de limbajul în care este scris pachetul grafic.

În structura multitasking a acestei implementări, interfața îndeplinește o funcție suplimentară, de codificare a comenzilor și parametrilor și de transmitere către nucleu.

La sincronizarea celor două procese - aplicația și

nucleul grafic - s-au luat în considerare:

- implicațiile implementării conceptului de aminare a unor acțiuni de modificare a imaginii;
- necesitatea optimizării transferului de date;
- asigurarea consistenței informațiilor furnizate de către nucleu sau circuitul de comandă device-driver, în raport cu funcțiile apelate anterior.

Se disting două categorii de funcții - de intrare și de interogare - care reclamă o sincronizare imediată și fac apel la lista GKS, listele stațiilor de lucru, lista segmentelor, precum și la starea spațiului de afișare la un moment dat.

De asemenea, sincronizarea imediată cu nucleul este solicitată de unele funcții de control al stațiilor de lucru care implică actualizarea imaginii și de valoarea "ASAP" (as soon as possible), atribuită parametrului de stare "DEFERAL STATE".

#### 4. Nucleul grafic

Nucleul de rutine grafice independente de dispozitiv este conceput ca un task lansat în execuție de programul de aplicație prin comanda OPEN GKS și eliberat prin comanda CLOSE GKS.

Task-ul gestionează lista GKS, lista segmentelor, memoria de segmente independente de stații - WISS, stațiile de lucru de tip metafisier - METAFILE INPUT, METAFILE OUTPUT.

Nucleul asigură funcțiile de interogare, mai puțin pe cele care apelează listele stațiilor de lucru și efectuează transformarea de normalizare.

Funcțiile de intrare și cele de setare a atributelor dinamice suportă o validare a stării stației de lucru, după care parametrii sînt transmiși task-ului circuitului de comandă (device-driver) respectiv.

Un set de rutine asigură comunicarea cu programul de aplicație, efectuînd recepția buffer-ului, decodificarea comenzilor și parametrilor, precum și apelul funcțiilor GKS.

#### 5. Stația grafică de lucru

În structura prezentată, nivelul inferior legat de dispozitivul grafic este asigurat de un task, care îndeplinește funcția de circuit de comandă (device-driver) pentru o stație de lucru particulară.

Task-ul este lansat în execuție de către nucleu prin comanda CLOSE WORK STATION.

Un set de rutine, similar celui prevăzut în nucleu, îndeplinește funcția de decodificare a comenzilor și parametrilor din zona de comunicație.

Funcțional, în sens GKS, în această implementarea circuitului de comandă (device-driver-ului) îi revin următoarele sarcini: gestionarea listei stației de lucru, actualizarea imaginii la stație, transformarea de

segment și transformarea stației de lucru, operația de decupare (clipping), afișarea primitivelor și gestionarea interacțiunii grafice.

Software-ul stației de lucru cuprinde o parte independentă de dispozitiv, dar dependentă de stația de lucru și o parte dependentă de dispozitiv.

Facilitățile diverse oferite de un echipament grafic, mergînd pînă la implementarea VLSI a unor funcții GKS, determină atît dimensiunile task-ului, cît și viteza de răspuns. Pentru echipamentele mai slabe, performanțele sînt reduse, în condițiile unui efort mare de realizare a lui, efort care include simularea la nivel software a unor funcții.

#### 6. Concluzii

Abordarea de tip multitasking a constituit soluția cea mai adecvată de implementare a standardului grafic în condițiile date, prezentînd totodată avantajul dezvoltării în mod independent a circuitelor de comandă (device-driver-elor) pentru noile echipamente.

În stadiul actual al implementării sînt realizate interfața FORTRAN, nucleul de rutine grafice independente de dispozitiv și circuitul de comandă (device-driver-ul) pentru clasa de echipamente TEKTRONIX produse în țară (DAF, DAFC, VDT52, VDT240, ALFAGRAF) și se află în curs de testare driverele pentru stația grafică GKS GT300 și masa de desenat KOVO.

Implementarea permite utilizarea simultană a trei stații de lucru grafice, oferind posibilitatea vizualizării pe trei ecrane, facilitate utilă în aplicațiile de proiectare asistată de calculator.

Numărul maxim de stații de lucru deschise simultan este șase, incluzînd memoria de segmente independentă de stații și două stații de tip metafisier.

#### Bibliografie

1. ENDERLE, G., KANSKI, K., PFAFF, G. - *Computer Graphics Programming: GKS - The Graphics Standard* Springer-Verlag, Berlin, 1987.
2. FOLEY, J.D., VAN DAM, A. - *Fundamentals of Interactiv Computer Graphics*, Addison-Wesley, Reading, 1982.
3. FOLEY, J.D., VAN DAM, A. - *Draft International Standard, Information Processing. Graphical Kernel System (GKS), Functional Description*, ISO/TC 97/SC5, ISO/DIS 7942.
4. HERMAN, I., REVCZKY, I., TOLNAY, T. - *A Concept for GKS Machine*, In: Proc. EUROGRAPHICS '85 - North Holland, Amsterdam, 1985.