Matjaž Debevc
Rajko Svečko
Mark Martinec*
Tehniška fakulteta Maribor
* Institut »Jožef Stefan«

UDK 681.326

POVZETEK

V želji, da bi uporabili Partner kot samostojen grafični terminal, je bil napisan grafični protokol med njim in med računalnikom VAX. S tem smo dosegli, da se uporabljajo vsi Partnerjevi grafični ukazi. Pri tem smo uporabili pomožni knjižnici TEKUSR na Partnerju in SUNLET na VAX sistemu.

ABSTRACT

In wish that we use Partner as an indepandet graphic terminal, was written a special graphic protocol between Partner and Vax. With that we are able to use all Partner's graphics instructions.
We used additional libraries: TEKUSR on Partner and SUNLET on VAX.

## 1. UVOD

Zaradi raznovrstnih možnosti uporabe vseh mogočih grafičnih paketov, ki so namenjeni velikim, dragim in težko dostopnim tujim grafičnim terminalom bi bilo ugodno, ko bi lahko vse te pakete uporabljali kar na mikroračunalnikih - sadovih naše, jugoslovanske ustvarjalnosti.

Primer takega mikroračunalnika je Iskra-Delta Partner z grafično kartico. Mikroračunalnik Partner je samostojna enota z enobarvnim, grafičnim, rastrskim terminalskim zaslonom in največjo resolucijo 1024 * 512. Partner bi zadostoval za osnovno spoznavanje delovanja grafike, vendar le do takrat, ko bi uporabnikove želje postale zahtevnejše. Žal Partner nima dovolj prostora za velike grafične pakete, ponavadi instalirane v HOST računalniku.

Takšni paketi so na primer GKS(Graphical Kernel System), CGI(Computer Graphics Interface), CGM(Computer Graphics Metafile), PHIGS(The Programmer's Hierarchical Interactive Graphics System). Za večino programskih paketov je značilno, da poskušajo biti čimbolj aparaturno neodvisni. Izjema je le krmilni del(DRIVER), ki skrbi za pravilno komunikacijo osrednjega neodvisnega dela s terminalom. Za vsak terminal ga je potrebno napisati znova. Da bi lahko napisali takšne programe za Partnerja, ga moramo uporabiti kot samostojen grafični terminal hkrati z HOST računalnikom, s katerim je povezan. Na ta način lahko kličemo grafične podprograme in funkcije v HOST računalniku in pri tem izkoriščamo vse grafične lastnosti mikroračunalnika Partner.

Za realizacijo te komunikacije je potreben protokol za čim hitrejši odtok in pritok informacij v mikroračunalnik.

## 2. PROTOKOL GRAFIČNE KOMUNIKACIJE

Protokol pomeni v splošnem dogovor o obnašanju komunikacije med računalniki oziroma terminali. Protokoli so odgovorni za generiranje in procesiranje podatkov. Najooli

znana mreža OSI/ISO( Open System Interconection / International Standard Organisation) na primer vsebuje sedem nivojev:

1. fizični nivo
2. nivo podatkovnih povezav
3. mrežni nivo
4. prenosni nivo
5. usklajevalni nivo
6. predstavitveni nivo
7. aplikacijski nivo

Za usklajevanje aplikacijskih nivojev uporabimo aplikacijski protokol; v tem okviru se giblje tudi protokol za grafično komunikacijo.

Osnovni in najbolj grobi princip protokola za grafično komunikacijo je v tem, da program v Partnerju prebere niz določenih znakov, poslanih s HOST računalnika, in pri tem pokliče ustrezni Partnerjev grafični ukaz. Kadar želimo risati na Partnerju brez pomoči HOST računalnika uporabljamo grafične ukaze s sistemske datoteke BIOLIB.PAS. V njej so podprogrami in funkcije za risanje na Partnerju. Pri tem je treba paziti le, da pišemo programe s katerimi kličemo te ukaze, v TURBO pascalu. Ukaze za komuniciranje z grafičnim procesorjem kličemo s pomočjo funkcije BIOS(29) - (Basic Input/Output System); od tod tudi ime datoteke.

S HOST računalnikom, v našem primeru z VAX-8800 in MikroVAX, želimo najti dostop do teh podprogramov in funkcij v datoteki BIOLIB.PAS v Partnerju.

Za doseg te realizacije, pošiljamo s HOST računalnika nek določen niz znakov.

Na Partnerju deluje program za zaznavanje znakov na vhodu/izhodu, pri spoznavnem nizu zaustavi nadaljno vnašanje in izvede ustrezno grafično funkcijo. Po potrebi pošlje tudi določen niz znakov nazaj k HOST računalniku.

## 3. REALIZACIJA PROGRAMA ZA PARTNER

Glavna naloga tega programa je torej, da bere niz znakov iz vhoda/izhoda v Partner, pri določenem spoznavnem nizu pokliče ustrezni grafični ukaz in po potrebi pošlje ustrezne parametre HOST računalniku.

Problem se pojavi pri branju in pisanju na vhod/izhod Partnerja, kjer je bilo potrebno tudi občasno zaustavljanje tega procesa. Ta problem je rešen s podprogrami in funkcijami, napisanimi v datoteki TEKUSR.INC. Ti nam omogočajo dokaj preprosto branje ali pisanje na vhod/izhod Partnerja. Razvili so jih strokovnjaki v Iskri - Delti v Ljubljani.

Pri iskanju spoznavnega niza grafičnih znakov ne smemo motiti ostalih nizov, ki označujejo ukaze, namenjene HOST računalniku. Poiskati je treba takšen niz znakov, ki se pri normalnih ukazih skorajda nikoli ne pojavi . Pri bolj zmogljivih grafičnih terminalih uporabljamo ubežne sekvence (escape sequence) za klicanje grafičnih funkcij. Partner tega ne pozna, zato smo se odločili za razpoznavni niz "%@". Znak "%" se zelo redko pojavi, vendar je še mnogo manjša verjetnost, da se bo za tem znakom pojavil znak "@". Pred vsakim nadaljnim ukazom stoji zopet znak "@" za kontrolo, če bi prišlo do kakšne motnje pri komunikaciji. S tem je dana možnost, da se ne zruši celoten sistem, ampak je napaka samo pri enem ukazu. Za znakom "@" sledi spoznavna koda ukaza.

S to kodo ugotovi program z imenom GRAFIKA, kateri ukaz bo moral poklicati iz datoteke BIOLIB.PAS. Tej kodi sledi niz znakov, ki predstavljajo parametre, potrebne za tekoči grafični ukaz. Niz parametrov se zaključi z znakom "@" za označitev začetka naslednjega ukaza. Za popolno zaključitev pošiljanje znakov grafike nam služi znak "Z".

Za izvajanje grafične komunikacije vtipkamo ukaz GRAFIKA, za zaključitev komunikacije uporabimo znak "<CTRL>\".

## 4. REALIZACIJA MODULA ZA VAX

Za pošiljanje in sprejemanje kod s HOST računalnika smo razvili modul z imenom GRAFPART.PAS. Modul nam omogoča klicanje ukazov z enakimi imeni, kakor pri BIOLIB.PAS v Partnerju tako, da pošlje ustrezne razpoznavne kode in njim pripadajoče parametre. Pred tem poskrbi za pretvorbo parametrov v znake.

Naravna števila se pretvorijo v niz številskih znakov in logične spremenljivke v posamezen znak (TRUE <=> "T"). Pri sprejemu pa se znaki pretvorijo v ustrezne vrednosti parametrov. V datoteki GRAFPART je spremenjeno le ime gtext v text, dodani pa so podprogrami InitWs, CloseWs in Update.

Za interaktiven vhod in izhod smo uporabili knjižnico SUNLET, točneje modul TTIO. Knjižnico so razvili na Institutu Jožef Stefan v Ljubljani.

Da bi bil komunikacijski program čim bolj učinkovit, se lahko držimo naslednjih pravil:

- poslane kode naj bodo čim krajše,
- za spoznavanje ukazov raje uporabimo črke namesto številk, s čimer prihranimo pomnilniški prostor,
- pri kodiranju oziroma pri dekodiranju uporabimo hitre algoritme,
- uporabimo knjižnico SUNLET-TTIO,

- za izpis znakov imamo tri možnosti:

a) uporaba ločila, ki se ne sme pojaviti v tekstu dvakrat zapored,
b) uporaba ločila, ki se v tekstu ne sme pojaviti,
c) uvedba parametra za število črk v nizu.

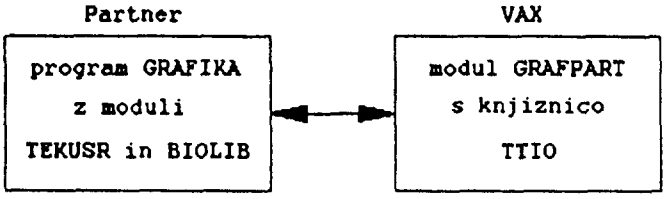
Za izpis teksta smo uporabili možnost pod c), ki je bila za to komunikacijo najbolj primerna. Naslednji primer nam prikaže pošiljanje kod s strani HOST računalnika (GRAFPART):

%@A1@C@H20 20 7GRAFIKA@B

| | |
|---|---|
| %@ | = začetna razpoznavna koda |
| A | = vklop grafike |
| 1 | = tip resolucije |
| @C | = brisanje zaslona |
| @H | = znak za tekst |
| 20 20 | = vrednosti za način pisanja |
| 7 | = število črk |
| GRAFIKA | = tekst |
| @B | = zaključek grafike |

Ko dobi program GRAFIKA na Partnerju na primer niz "@A1", pokliče funkcijo ginit(1) z datoteke BIOLIB.PAS.

## 5. SHEMA PROGRAMOV:




## 6. SEZNAM UKAZOV IN NJIHOVO DELOVANJE

| | |
|---|---|
| InitWS - | inicializacija delovne postaje in knjižnice SUNLET |
| CloseWS - | zaključitev dela z delovno postajo |
| ginit - | postavitev v grafični način in nastavitev resolucije |
| gexit - | zaključitev grafičnega načina |
| gclr - | brisanje grafičnega načina |
| gxy - | premik na pozicijo (x,y) |
| text - | pisanje grafičnega teksta |
| vbar - | nariše blok |
| circ - | nariše krog |
| disc - | nariše polni krog |
| ring - | nariše zapolnjen kolobar |
| draw - | nariše črto |
| vect - | nariše črto z relat. koordinatami |
| scroll - | premik slike po y-osi |
| getpix - | pove ali je točka na mestu, kjer se nahaja grafični kurzor prižgana |
| qfill - | polnjenje omejenih področij v vodoravni ali navpični smeri |
| getcursor - | vrne koordinate točke |
| Hardcopy - | kopira ekran na printer |
| Update - | ažuriranje slike |

## 7. PRIMER

Naslednji primer nam prikaže uporabo podprogramov, napisanih na računalniškem sistemu VAX-8800.

Nariši poln krog s koordinatama x = 100 in y = 100 in s polmerom rad = 50 !

```
[INHERIT('grafpart')]
      { vstavitev ukazov za grafiko }
program primer;
begin
{ inicial. delovne postaje in SUNLET }
  InitWs;
{ inicializacija grafike 1024 * 512 }
  ginit(1);
{ risanje polnega kroga }
  disc(100,100,50);
{ zaključitev grafičnega načina }
  gexit;
{ zaključitev knjižnice SUNLET }
  CloseWs;
end.
```

Ko program zaključimo, ga prevedemo in nato povežemo z objektno datoteko GRAFPART.OBJ in knjižnico SUNLET.

## 8. ZAKLJUCEK

Ta komunikacija nam torej omogoča pisanje krmilnih programov za grafične pakete tako, da uporabljamo ukaze, ki jih pozna mikroračunalnik Partner. Seveda ne smemo pozabiti, da mora pri tem na Partnerju ves čas delovati komunikacijski program z imenom GRAFIKA.

Programi so napisani brez odvečnih znakov in poskušajo omogočiti čim hitrejšo komunikacijo med HOST računalnikom(VAX 8800, MikroVAX) in mikroračunalnikom Partner. Razlika v hitrosti delovanja grafičnega programa na samem Partnerju in hitrosti delovanja programa preko HOST računalnika je skorajda neopazna.

Knjižnico je uporabljena tudi za pisanje krmilnega programa GKS sistema za grafiko na Partnerju.

## LITERATURA:

TURBO PASCAL,
Gams Matjaž:
Osnove dobrega programiranja,
VAX - 11 PASCAL:
Language Reference Manual,
VTO ERI:
Projekt-Mreže.