

F E L A N 4

P Ř Í R U Č K A
P R O G R A M Á T O R A

1. Programové rozhraní sítě FELAN4

Popisované rozhraní poskytované komunikačním ovladačem sítě FELAN4 dovoluje aplikačním programům přístup ke službám sítě a tím k přenosu zpráv mezi jednotlivými počítači zapojenými v síti.

1.1 Typy zpráv

Uživatelům dostupné operace umožňují vyslat a přijmout zprávu. Existuje 256 druhů (typů) zpráv. Většina zpráv je vyhrazena pro vyšší vrstvy programového vybavení FELAN. Zprávy 160 (0A0H) až 239 (0EFH) mohou využívat ostatní tvůrci síťových programů.

Bylo by vhodné, aby tito tvůrci oznamovali a konzultovali s autory FELANu, které z typů zpráv používají. Lze tak zabránit konfliktním situacím při provozování síťových programů vzniklých na různých pracovištích a využívajících stejné typy zpráv.

1.2 Volání služeb přenosu zpráv

Služby vysílání a příjmu zpráv poskytuje komunikační modul prostřednictvím programového přerušení 48H. Pro vyslání zprávy je určena funkce 0 tohoto přerušení. Pro příjem se využívají funkce 10H až 12H. Tyto funkce využívají registry AX, BX, EX, CY. Zbývající registry jsou nezměněny. Jiné funkce tohoto přerušení a přerušení 49H jsou rezervována pro jiné účely.

1.3 Vysílání zpráv

Požadavek na vysílání je předán komunikačnímu modulu. Po zkontrolování formální správnosti požadavku je zařazen do vysílací fronty a řízení je předáno zpět uživatelskému programu.

Vstup: AH 0
 ES:BX Adresa SCB - Send Control Block
 SCB obsahuje informace nutné k vyslání zprávy

Výstup: AL Odpověď
 CY Nastaven při chybě

Send Control Block obsahuje tyto informace:

+0 S_REPLY Byte odpovědi. První odpověď je nastavena po vznesení požadavku na vysílání (chyba nebo "dosud se vysílá"). Tato hodnota je k dispozici i v registru AL. Po ukončení vysílání je zde druhá odpověď informující o úspěšnosti vysílání. V tomto bytu tedy uživatelský program sleduje průběh vysílání.

+1 S_RESERVE Rezervováno pro vnitřní potřebu.

- +2 S_RQACK Má-li tento byte nenulovou hodnotu, adresát potvrzuje přijetí zprávy. V takovém případě příkaz vyšší zprávu končí přijetím potvrzení nebo vyčerpáním časového limitu, ve kterém měl potvrzení obdržet. Časový limit začíná okamžikem fyzického odeslání zprávy, uživatel jeho velikost nemůže ovlivnit.

- +3 S_TMOUT Time-out pro vysílání. Požadavky na vysílání jsou odebírány z vysílací fronty a zprávy vysílány. Je-li síť přetížena a zprávu se nedaří vyslat, je po zvolené době (time-out) hlášena chyba. Velikost time-outu se udává v násobcích 55 ms.
0 - jádro použije standardní hodnotu 3
>0 - toto je time-out
<0 - vysílání bez časového omezení

- +4 S_LANTO Určeno pro adresaci jiných sítí. Je zde uvedeno číslo sítě, nula znamená, že adresát pracuje ve stejné síti jako odesílatel zprávy. Doporučujeme nastavit nulu.

- +5 S_ADRTO Adresát

- +6 S_LEN Délka vysílané zprávy je uložena ve slově. Vysílaná zpráva je dlouhá max. 591 bytů.

- +8 S_BUF Dvouslovo obsahující adresu bufferu s textem zprávy. Text zprávy začíná bytem s typem (druhem) zprávy.

1.4 Příjem zpráv

Pro přijímání zpráv jsou určeny funkce:

- 10H požadavek na trvalý příjem zvolených typů zpráv
- 11H vlastní příjem zprávy
- 12H zrušení trvalého příjmu zpráv

Komunikace mezi uživatelským programem a komunikačním modulem probíhá prostřednictvím datové struktury Receive Control Block - RCB. Ve funkcích 11 a 12 je nutno uvádět tu adresu RCB, která byla použita pro funkci 10. V RCB jsou obsaženy položky:

- +0 R_REPLY Byte odpovědi na prováděné příkazy.

- +1 R_RESERVE Vyhrazeno pro vnitřní potřebu.

- +2 R_MIN

- +3 R_MAX Byty R_MIN a R_MAX udávají rozsah typů (čísel) přijímaných zpráv.

- +4 R_RCVD Hodnota položky udává počet zpráv, které byly

- přijaty a čekají v přijímací frontě na předání uživateli.
- +5 R_TMOUT Time-out pro příjem. Pokud po vydání příkazu "přijmi zprávu" není zpráva v přijímací frontě a ani není přijata v zadaném časovém limitu, je to považováno za chybu. R_TMOUT se udává v násobcích 55 ms:
- 0 - jádro použije standardní hodnotu 3
 - >0 - toto je time-out
 - <0 - příjem zprávy bez časového omezení
- +6 R_LANFROM Adresace jiných větví sítě. Je zde uvedeno číslo sítě odesilatele zprávy. Nula znamená, že odesílatel pracuje ve stejné síti jako příjemce.
- +7 R_ADRFROM Adresa odesilatele zprávy.
- +8 R_LEN Slovo obsahující délku bufferu pro uložení zprávy. Lze přijmout zprávu dlouhou nejvýše 591 bytů.
- +10 R_BUF Dvouslovo udává adresu bufferu s přijatou zprávou. Prvním bytem ve zprávě je její typ.

Funkce 10H - požadavek na příjem zpráv

Vstup: AH 10H
 ES:BX Adresa RCB
 V RCB jsou nastaveny položky R_MIN a R_MAX

Výstup AL Byte odpovědi
 CY Nastaven při chybě

Touto funkcí uživatelský program sděluje komunikačnímu modulu, že chce trvale přijímat zprávy, jejichž typ je z intervalu R_MIN až R_MAX.

Funkce 11H - přijmi zprávu

Vstup AH 11H
 ES:BX Adresa RCB
 V RCB musí být nastaveny položky R_TMOUT, R_LEN a R_BUF

Výstup AL Byte odpovědi
 CY Nastaven při chybě
 R_LEN Udává skutečnou délku přijaté zprávy

Uživatelskému programu je z přijímací fronty předána přijatá zpráva. Není-li ve frontě vhodná zpráva, čeká se po dobu R_TMOUT na její přijetí ze sítě. Po tuto dobu je uživatelský program zastaven. Před provedením služby uživatel v R_LEN nastavení max. délka zprávy (velikost bufferu), při předání zprávy

uživatelskému programu je do R_LEN zapsána skutečná délka přijaté zprávy.

Funkce 12H - zruš příjem zpráv

Vstup	AH	12H
	ES:BX	Adresa RCB
Výstup	AL	Byte odpovědi
	CY	Nastaven při chybě

Tuto funkci musí program použít, jestliže již nechce dále přijímat žádné zprávy nebo se chystá ukončit svoji činnost. Není-li funkce použita, komunikační modul i nadále přijímá požadované zprávy a aktualizuje položku R_RCVD v RCB bloku.

1.5 Zjištění vlastní adresy

Funkce 02H - zjištění vlastní adresy

Vstup	AH	02H
Výstup	CH	číslo větve sítě
	CL	adresa uzlu v rámci větve sítě
	ES:BX	Adresa devítiznakového řetězce, který udává jméno uzlu (např. FELAN0001)

Tato služba nevyvolá žádnou komunikaci po síti. Pouze zjistí od komunikačního ovladače síťovou adresu uzlu a jeho jméno.

1.6 Chyby při příjmu a vysílání

K chybám dochází vlivem vznesení chybných požadavků na přenos zpráv, vlastnostmi použitého protokolu CSMA/CD nebo dosažením maxima možností FELANu.

- 1 Příkaz se provádí
- 0 Příkaz se úspěšně provedl
- 1 Formální chyba požadavku
- 2 Požadovaný typ zpráv již někdo přijímá
- 3 Příliš mnoho požadavků na trvalý příjem
- 4 Adresa RCB nebyla použita ve funkci 11
- 5 V SCB je uvedena chybná délka zprávy
- 6 Příliš mnoho požadavků na příjem
- 7 Vyžádané potvrzení nepřišlo v časovém limitu
- 8 Přijatá zpráva je delší než je uvedeno v R_LEN, zpráva je oříznuta
- 9 Time-out

Používání uvedených funkcí vyžaduje velkou zodpovědnost programátora. Autoři FELANu proto dávají k dispozici dvě různé jednotky pascalských knihovních modulů, LANLIB.PAS a USERLAN.PAS. Druhá z uvedených knihoven sice poněkud omezuje

autora aplikačních programů, ale naproti tomu mu zaručuje, že jeho programy nebudou rušit provoz na síti. Jednotky jsou dodávány i ve zdrojovém jazyce, protože jednotky TPU nejsou přenositelné mezi jednotlivými verzemi Turbopascalu. Autoři FELANu se domnívají, že knihovny budou používat pouze zkušení programátoři, pro něž bude dostatečným návodem přečtení zdrojového textu jednotky. K tomu může napomoci přiložený ukázkový program BRO.PAS.

1.7 Ukázka použití jednotky USERLAN.PAS

```
unit userlan;
{ Uživatelské rozhraní komunikačního ovladače sítě FELAN      }
{ verze 3.11.1990                                           J.Kašpar      }
interface
type
buffer = record
    address : word ;                               { adresa uzlu }
    length : word ;                               { délka paketu včetně typu }
    messagetype : byte ;                          { typ paketu }
    message : array [1..590] of byte ;            { data }
end ;

function MYaddress : word ;                       { dodá adresu vlastního uzlu }

function OPENLINE ( min,max : byte ) : integer ;
    { otevření kanálu - povolí příjem paketů min..max}
    { 0 -> O.K.
      1 -> chybná čísla typu paketu
      2 -> požadované typy paketů již někdo přijímá
      3 -> příliš mnoho žádostí o příjem
      7 -> druhé volání OPENLINE bez předchozího
          CLOSELINE }

function RECEIVED : BOOLEAN ;
    { je paket přijat ale dosud neodebrán? }

function RECEIVE ( var buf : buffer ) : integer ;
    { odebrání přijatého paketu }
    { 0 -> O.K.
      4 -> nebyla provedena funkce OPENLINE
      8 -> přijatá zpráva je delší než
          buffer - byla oříznuta
      9 -> timeout }

procedure CLOSELINE ;                            { uzavření kanálu - zrušení příjmu }
function SEND ( var buf : buffer ) : integer ;   { vyslání paketu }
    { 0 -> O.K.
      1 -> chybný typ paketu
      4 -> nebyla provedena funkce OPENLINE
      5 -> chybná délka paketu
      6 -> chybný adresát }

function BROADCAST ( var buf : buffer ) : integer ;
    { vyslání paketu všem }
    { 0 -> O.K.
      1 -> chybný typ paketu
      4 -> nebyla provedena funkce OPENLINE
      5 -> chybná délka paketu }
```

```

program broadcasting(output) ;
uses crt, userlan ;
const
  msgnumber = 239 ;           { číslo používané zprávy }
var
  rb,sb : buffer ; { vyrovnávací paměti pro příjem a vysílání }
  stop : boolean ;
  k : char ;
  i,e : integer ;
begin
  writeln ;
  writeln('FELAN : BROADCAST           (C) JK 1990 ') ;
  writeln('NODE : ',MYaddress) ;
  writeln ;
  writeln('Help :') ;
  writeln ;
  writeln('<RETURN> -> send message') ;
  writeln(' <ESC>   -> end of run') ;
  writeln ;
  writeln ;
  write('BRO> ') ;           { prompt }
  sb.length := 1 ;         { vyprázdnění vysílacího bufferu }
  sb.messageType := msgnumber ; { typ (číslo) odesílaných zpráv }
  stop := false ;
  e := openline ( msgnumber,msgnumber ) ;
                                { chceme přijímat zprávy č.250 }

  if e <> 0 then
    writeln('OPENLINE ERROR ',E) ;
  while not stop do
    begin
      if keypressed then
        begin
          { byla stisknuta klávesa }
          k := readkey ;           { kód stisknuté klávesy }
          if k = chr(0) then k := readkey { speciální klávesa }
          else                       { obyčejná klávesa }
            with sb do
              begin
                if length > 560
                  then k := chr($0d) ; { konec bufferu - jako return }
                if ord(k) in [0..$1f] then { nezobrazitelný znak }
                  case ord(k) of
                    $0d: begin { return -> odeslání zprávy }
                          writeln ;
                          e := BROADCAST(sb) ;
                          { vysílej zprávu pro všechny }
                          if e <> 0 { došlo k chybě při vysílání ? }
                            then writeln('BROADCAST ERROR ',E) { ano }
                            else writeln('MESSAGE SENT') ;      { ne }
                          length := 1 ; { vyprázdnění vysílacího bufferu }
                          write('BRO> ') ; { prompt }
                        end ;
                    $1B: stop := true ; { ESC -> konec }
                    else ;
                  end
                else
                  begin

```



```

        write(k) ;                               { echo na obrazovku }
        message[length] := ord(k) ;
        length := length + 1 ; { přidání znaku na konec zprávy }
    end ;
end ;
if received then                                { něco nám přišlo }
begin
    e := receive(rb) ;                          { přečtení zprávy }
    if e <> 0 { došlo k chybě při čtení zprávy ? }
    then writeln('RECEIVE ERROR ',E)           { ano }
    else                                       { bez chyb }
    with rb do
        if address <> MYaddress then
            { mnou odeslané zprávy se nevypisují }
        begin
            writeln ;
            writeln('MESSAGE FROM NODE ',address,':') ;
            for i := 1 to length - 1 do
                write(chr(message[i])) ;
            writeln ; { opis přijaté zprávy na obrazovku }
            write('BRO> ') ; { prompt }
            if sb.length <> 1 then { obnovení rozepsané zprávy }
            for i := 1 to sb.length - 1 do
                write(chr(sb.message[i])) ;
            end ;
        end ;
    end ;
end ;
closeline ; { ukončení příjmu zpráv }
end.

```

1.	Programové rozhraní sítě FELAN	2
1.1	Typy zpráv	2
1.2	Volání služeb přenosu zpráv	2
1.3	Vysílání zpráv	2
1.4	Příjem zpráv	3
1.5	Zjištění vlastní adresy	5
1.6	Chyby při příjmu a vysílání	5
1.7	Ukázka použití jednotky USERLAN.PAS	7