CrtWin

JEDNOTKA PRO PRÁCI S OBRAZOVKOU A KLÁVESNICÍ

Příručka uživatele a programátora

***SofCon*®spol. s r.o.**
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: http://www.sofcon.cz

Verze dokumentu 1.11

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.
SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsané. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.
SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 10.06.2003
Datum posledního uložení dokumentu: 10.06.2003
(Datum vydání a posledního uložení dokumentu musí být stejné)

Obsah :

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.XX	We		První vydání.
1.10	1.XX	Tu	22.05.2003	Úprava dokumentu dle ISO9000.
1.11	2.XX	Tu	10.06.2003	Přidaná procedura CrtClose.

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotky pro práci s obrazovkou a klávesnicí.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu není potřeba číst žádný další manuál, ale je potřeba se orientovat v používání programového vybavení SofCon.

Popis formátu verze knihovny a souvisejících funkcí je popsán v manuálu LibVer.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu Termíny a definice.

3. Úvod

Jednotka CrtWin byla vytvořena pro zjednodušení práce uživatele při přístupu paralelních procesů k obrazovce a klávesnici počítače třídy PC. Programátor při práci s obrazovkou může využívat standardní služby pro práci v alfanumerickém režimu poskytované překladačem. Před jejich použitím je však nutné, aby proces, voláním služby jednotky CrtWin, získal výhradní právo přístupu k obrazovce a klávesnici. Jednotka zároveň poskytuje prostředky pro vytváření pracovních ploch, tzv. oken, které práci s obrazovkou značně usnadní.

4. Popis konstant a typů

```
cVerNo = např. $0251; { BCD formát }
cVer   = např. '02.51,07.08.2003';
```

Číslo verze jednotky v BCD tvaru a v textové podobě včetně datumu změny.

```
TRect = record
          X,Y,W,H: byte;
end;
```

TRect obsahuje definici umístění a velikosti okénka. **X,Y** je souřadnice levého horního rohu, **W** je šířka a **H** je výška okénka.

```
TFrame = array[0..7] of Char;
```

TFrame jsou znaky pro orámování okénka. Zadáváme znaky rohů a stran pro rámování v pořadí od levého horního rohu ve směru pohybu hodinových ručiček.

```
FrameS : TFrame=('┌','─','┐','│','┘','─','└','│');
```

FrameS je konstanta pro jednoduché orámování okna.

```
FrameD : TFrame=('╔','═','╗','║','╝','═','╚','║');
```

FrameD je konstanta pro dvojité orámování okna je stejná jako **FrameS**, ale obsahuje symboly pro dvojitý rámeček. Poněvadž jsme neměli fonty do tiskárny s dvojitým rámečkem, jsou uvedeny pouze jednoduché rámečky.

5. Popis objektu TCrtWin

```
PCrtWin = ^TCrtWin;
TCrtWin = object
```

Objekt umožňuje vytvářet na obrazovce pracovní plochy, tzv. okna. Při výpisu na takto vytvořená okna je nutno dodržet následující postup. Nejprve je třeba voláním metody **LockWin** získat výhradní právo přístupu k obrazovým službám a k danému oknu. Poté lze provést výstup na obrazovku, změnu parametrů pro výstup na obrazovku nebo vstup z klávesnice. (**Write, WriteLn, GotoXY, WhereX, WhereY, TextColor, TextBackground, HighVideo, LowVideo, NormVideo, KeyPressed, ReadKey, ClrEol, ClrScr, DelLine, Window**). Na konec je třeba voláním metody **UnLockWin** zrušit přístup k obrazovým službám a k danému oknu. Metody **LockWin** a **UnLockWin** je možné do sebe vnořovat. Zároveň je možné, aby vnořené metody byly od různých instancí (oken).

5.1. Pole

```
wRect: TRect;
```

wRect obsahuje údaje o umístění a velikosti vytvořeného okna. Každé okno je umístěno absolutně vzhledem k základní ploše **MainWin**. Zadává se X,Y,W,H: byte; X,Y=souřadnice levého horního rohu, W=šířka, H=výška. Proměnná je inicializována parametrem Rect konstruktoru Init.

```
wX,wY: Byte;
```

wX,wY jsou souřadnice umístění kurzoru v daném okně. Po vytvoření okna je kurzor nastaven na pozici 1, 1. Proměnné jsou aktualizovány při volání metody **UnLockWin**.

```
wA: Byte;
```

wA je údaj o hodnotě atributu použitého při výpisu v daném okně. Tento atribut nese údaje o barvě pozadí textu i textu samotného. Tato informace je uložena v jednom byte následovně: B,PPP,TTTT. B je příznak blikání textu (B=1 znamená, že text bude blikat). PPP jsou tři bity nastavující barvu pozadí textu a TTTT jsou čtyři bity označující barvu textu. Pokud pomocí služby 3 obrazového přerušení 10 zakážeme blikání textu, lze pro barvu pozadí použít i bit B. Přiřazení barev je následující:

0 černá
1 modrá
3 blankytně modrá
4 červená
5 fialová
6 hnědá
7 světle šedá
8 tmavě šedá
9 modrá přisvětlená
10 zelená přisvětlená
11 blankytně modrá přisvětlená
12 červená přisvětlená
13 fialová přisvětlená
14 žlutá
15 bílá

Po vytvoření okna je proměnná inicializována hodnotou **AW** konstruktoru **Init** a je aktualizována při volání metody **UnLockWin**.

```
wL1,wL2: Byte;
```

wL1,wL2 obsahují údaje o velikosti kurzoru použitého v daném okně. Po vytvoření okna je velikost nastavena parametry **CL1** a **CL2**. **CL1** určuje počáteční linku, **CL2** určuje koncovou linku kursoru. Proměnné jsou aktualizovány voláním metody **SetCursor** a **HiddenCursor**. Obsahují-li proměnné hodnoty $20, 0 je kurzor neviditelný.

5.2. Metody

5.2.1. Init

```
constructor Init(Rect: TRect; S: TString; Frame: TFrame;
                 AS, AR, AW: Byte; CL1, CL2: Byte);
```

Voláním konstruktoru **Init** vytvoříme nové okno dané velikosti s daným umístěním na obrazovce. **TRect** určuje umístění a velikost vytvářeného okna. **S** označuje nadpis vytvářeného okna. Nadpis je umístěn uprostřed horního okraje okna. **Frame** určuje jakým způsobem bude vytvářené okno orámováno. Jsou předdefinovány konstanty **FrameD** a **FrameS**. Pro orámování dvojitou čárou poslouží konstanta **FrameD**, pro orámování jednoduchou čarou konstanta **FrameS**. **AS** je hodnota atributu použitá při výpisu nadpisu S. **AR** je hodnota atributu použitá při orámování okna. Je-li AS = AR = 0 nebude vytvářené okno orámováno. **AW** je hodnota atributu použitá při výpisu v daném okně. Význam těchto atributů je popsán u proměnné **wA**. **CL1** a **CL2** určují velikost kurzoru použitého v daném okně. **CL1** určuje počáteční linku, **CL2** určuje koncovou linku kursoru. Jsou-li hodnoty $20, 0 nebude kurzor viditelný.

5.2.2. Done

```
destructor Done;
```

Destruktor **Done** slouží pro zrušení objektu daného okna.

5.2.3. LockWin

```
procedure LockWin;
```

Voláním metody je získáno výhradní právo pro přístup k zobrazovacím službám. Současně se dané okno stane oknem aktuálním. Uvnitř metody **LockWin** je volající proces pozastaven do chvíle než získá výhradní přístup k obrazovým službám. Při následném volání procedur **Write** a **WriteLn** bude výpis proveden právě do okna určeného danou instancí objektu **tCrtWin**. Při čekání na znak z klávesnice, při volání procedur **Read** a **ReadLn**, je při čekání na znak uvolněn přístup k obrazovým službám pro jiné procesy. Po stisku klávesy je tento přístup opět získán. Uživatel musí zajistit, aby klávesnicové služby volal zároveň jen jeden proces.

5.2.4. UnLockWin

```
procedure UnLockWin;
```

Metoda obnoví okno, které bylo aktivní před voláním odpovídající metody **LockWin** a případně uvolní přístup k zobrazovacím službám. Metody **LockWin** a **UnLockWin** je možné vnořovat do sebe. Lze vnořovat metody i od různých instancí objektu (oken).

5.2.5. PlaceFrame

```
procedure PlaceFrame(Rect: TRect; S: TString;
                     Frame: TFrame; AS, AR: Byte);
```

Metoda orámuje plochu, jejíž velikost a umístění určuje parametr **Rect**. Frame určuje znaky pro orámování. Jsou předdefinovány konstanty **FrameD** a **FrameS**. Pro orámování dvojitou čárou poslouží konstanta FrameD, pro orámování jednoduchou čarou konstanta **FrameS**. **AS** je hodnota atributu použitá při výpisu nadpisu S. **AR** je hodnota atributu použitá při orámování okna.

5.2.6. SetCursor

```
procedure SetCursor(CL1, CL2: Byte);
```

Metoda pro nastavení velikosti kurzoru v daném okně. Lze ji volat i bez použití metod **LockWin** a **UnLockWin**. **CL1** a **CL2** určuje velikost kurzoru použitého v daném okně. **CL1** určuje počáteční linku, **CL2** určuje koncovou linku. Jsou-li hodnoty $20, 0 nebude kurzor viditelný.

5.2.7. HiddenCursor

```
procedure HiddenCursor;
```

Voláním metody je potlačen výpis kurzoru v daném okně.

5.2.8. Save

```
procedure Save;
```

Metoda slouží jen pro vnitřní použití.

5.2.9. Restore

```
procedure Restore;
```

Metoda slouží jen pro vnitřní použití.

6. Popis proměnných

```
MainWin : PCrtWin=nil;
```

MainWin je ukazatel na instanci objektu **tCrtWin** vzniklou při volání procedury **CrtInit**. Popisuje hlavní okno na obrazovce.

```
ActWin  : PCrtWin=nil;
```

ActWin je ukazatel na aktuální okno na obrazovce.

```
WaitKey : Word=2;
```

WaitKey je počet wait stavů jádra Retos při čekání na stisk klávesy.

7. Popis procedur

7.1. CrtInit

```
procedure CrtInit
```

Procedura slouží k inicializaci jednotky CrtWin a k vytvoření hlavní pracovní plochy o velikosti odpovídající maximální velikosti obrazovky aktuálního obrazovému režimu. Na tuto plochu je možné se odkazovat prostřednictvím instance MainWin. Proceduru **CrtInit** je možné volat jen při spuštěném jádře operačního systému Retos. Tuto proceduru je nutné zavolat před vytvořením instancí objektu **tCrtWin**.

7.2. CrtClose

```
Procedure CrtClose
```

Procedura zruší všechna vytvořená okna, zruší instanci MainWin a uvolní paměť.

8. Příklad použití CrtWin

Ukázkový program vytváří dva různé typy procesů. Proces P1 vypisuje opakovaně na obrazovku hodnotu čísla i, kterou zároveň zvětšuje. Proces P2 čte pomocí funkce **ReadLn** znaky z klávesnice. Hlavní program vytvoří tři instance procesu P1 a jednu instanci procesu P2. Celkem jsou tedy spuštěny čtyři procesy, které se střídají při přístupu k obrazovým službám.

```
program TestCrtWin;
uses
  Kernel,
  CrtWin;

const
  MaxProc1=3;
  Rect1 : array[1..MaxProc1] of TRect=(
            (X: 2; Y: 2; W: 38; H: 10),
            (X: 42; Y: 2; W: 38; H: 10),
            (X: 2; Y: 14; W: 38; H: 10));

  Rect2 : TRect = (X: 42; Y: 14; W: 38; H: 10);

const
  FlEnd : Boolean=false;

{$F+} { proces výpisu na obrazovku }
procedure P1(Rect: TRect; S: String; AS, AR, AW: Byte);
{$F-}
var
  Win : PCrtWin;
  i   : Word;
begin
  Start('Proc1',8000,90,254);Exit;
 { vytvoření okna }
  Win:=New(PCrtWin,Init(Rect,S,FrameD,AS,AR,AW,$20,0));
  repeat
```

```
    Win^.LockWin;    { získání práva přístupu na obrazovku }
    WriteLn(i);      { výpis na obrazovku }
    Win^.UnLockWin;  { ukončení práva přístupu na obrazovku }
    ...
    {$R-}
    Inc(i);
    {$R+}
    Wait(2);
  until FlEnd;
end;

{$F+} { proces vstupu z klávesnice }
procedure P2(Rect: TRect; S: String; AS, AR, AW: Byte);
{$F-}
var
  Win : PCrtWin;
  SS  : String;
begin
  Start('Proc1',8000,90,254);Exit;
  { vytvoření okna }
  Win:=New(PCrtWin,Init(Rect,S,FrameD,AS,AR,AW,{ $20,0}1,8));
  repeat
    Win^.LockWin;       { získání práva přístupu na obrazovku }
    Write('Zadej : '); { výpis na obrazovku }
    ReadLn(SS);         { vstup z klávesnice }
    Win^.UnLockWin ;    { ukončení práva přístupu na obrazovku }
    ...
  until FlEnd;
end;

var
  i : Byte;
begin
  StartMain(250,254);
  InitInterruptStack(1,8000);
  StartTimeSlicing($8);

  CrtInit;
  MainWin^.HiddenCursor;

  for i:=1 to MaxProc1 do
    P1(ARect[i],'Proces',$7,$7,$7);      { spuštění procesu P1 }

  P2(ARect[MaxProc],'Proces',$7,$7,$7); { spuštění procesu P2 }

  SetPriority(10,254);
  repeat
  until FlEnd;
  Abort('*');
  TerminateMain;
end.
```