

CrtWin, CrtWin2

JEDNOTKY 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>

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 popsáno. 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: 17.02.2004

Datum posledního uložení dokumentu: 17.02.2004

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.	O dokumentu	5
1.1.	Revize dokumentu	5
1.2.	Účel dokumentu	5
1.3.	Rozsah platnosti	5
1.4.	Související dokumenty	5
2.	Termíny a definice	5
3.	Úvod	6
4.	Popis konstant a typů	6
5.	Popis objektu TCrtWin	7
5.1.	Položky	7
5.2.	Metody	8
5.2.1.	Init	8
5.2.2.	InitNoFrame	9
5.2.3.	Done	9
5.2.4.	LockWin	9
5.2.5.	UnLockWin	9
5.2.6.	PlaceFrame	9
5.2.7.	SetCursor	9
5.2.8.	HiddenCursor	10
6.	Popis proměnných	10
7.	Popis procedur	10
7.1.	CrtInit	10
7.2.	CrtClose	10
8.	Příklad použití CrtWin	10

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze CrtWin	Verze CrtWin2	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.
01.20	3.XX	2.XX	Wil	17.02.2004	Změna procedur Save a Restore na privátní, tj. v aplikaci se nesmí volat. Přidán popis rozdílů CrtWin a CrtWin2.

1.2. Účel dokumentu

Tento dokument slouží jako popis jednotek 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

Při použití paralelních procesů operačního systému reálného času Retos a při přístupu k obrazovce či klávesnici z více procesů pomocí standardních funkcí (např. Write, Read, ReadKey ...) a dalších funkcí definovaných jednotkou Crt jazyka Pascal, není zaručena reentrantnost. To znamená, že pokud například dva (či více) procesů ve stejný okamžik začnou vypisovat na obrazovku do určitého místa, nebude výpis konzistentní (navzájem si přenastavují pozice kurzoru, barvy výpisu apod).

Proto pro zjednodušení práce uživatele při přístupu paralelních procesů k obrazovce a klávesnici počítače třídy PC byla vytvořena knihovna **CrtWin**, která definuje objekt pro přístup k oknům obrazovky v textovém režimu a klávesnici. (Pozn: Přístup k obrazovce v grafickém režimu definuje knihovna **GraphWin**, viz příslušný manuál).

Programátor při práci s obrazovkou v textovém režimu může využívat většinu standardních služeb pro práci v alfanumerickém režimu poskytované standardní jednotkou Crt a System jazyka Pascal. Před jejich použitím je však nutné, aby paralelní proces, voláním příslušné 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 a zpřehlední.

Pokud daná aplikace nepoužívá systém reálného času o.s.Retos, nelze knihovnu **CrtWin** použít. Pokud ale programátor aplikace uvažuje o budoucím využití o.s.Retos, může si již výpisy na obrazovku připravit pomocí knihovny **CrtWin2**. Tato knihovna definuje identický interface pro použití v aplikaci. Rozdíl knihovny CrtWin2 oproti CrtWin je v tom, že CrtWin2 ve svých vnitřních implementacích nepoužívá služeb o.s.Retos.

Následující popis se týká obou knihoven **CrtWin** a **CrtWin2**, na případné rozdíly bude upozorněno.

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. Konkrétní hodnoty těchto konstant se mohou od tohoto dokumentu lišit.

```
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=('P','-', 'q','|','@','-', 'u','|');
```

FrameD je konstanta pro dvojité orámování okna.

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. Lze používat téměř všech standardních funkcí jednotky Crt (**Write**, **WriteLn**, **GotoXY**, **WhereX**, **WhereY**, **TextColor**, **TextBackground**, **HighVideo**, **LowVideo**, **NormVideo**, **KeyPressed**, **ReadKey**, **ClrEol**, **ClrScr**, **DelLine**, **InsLine**). Nelze používat funkci **Window**. Její použití by kolidovalo s okny na obrazovce, které definuje objekt TCrtWin. Po ukončení přístupu k obrazovce či klávesnici 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. Položky

```
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. Souřadnice na obrazovce jsou počítány od hodnoty 1. Položka je inicializována parametrem Rect konstrukturu.

```
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 barvy 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 Int10h zakážeme blikání textu, lze pro barvu pozadí použít i bit B.

Po vytvoření okna je položka inicializována parametrem **AW** konstrukturu a je aktualizována při volání metody **UnLockWin**.

Přiřazení barev odpovídá standardním konstantám jednotky Crt:

Tmavé barvy		Světlé barvy	
Hodnota	Barva	Hodnota	Barva
0	černá	8	tmavě šedá
1	tmavě modrá	9	světle modrá
2	tmavě zelená	10	světle zelená
3	blankytně modrá	11	světle blankytně modrá
4	tmavě červená	12	světle červená
5	fialová	13	růžová
6	hnědá	14	žlutá
7	světle šedá	15	bílá

wL1, wL2: Byte;

wL1, wL2 obsahují údaje o velikosti (tvaru) kurzoru použitého v daném okně. Po vytvoření okna je velikost nastavena parametry **CL1** a **CL2**. **CL1** určuje počáteční linku, **CL2** určuje koncovou linku kurzoru (počítáno od horního okraje znaku). Položky 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; const S: TString; const Frame: TFrame;
                AS, AR, AW: Byte; CL1, CL2: Byte);
```

Voláním konstrukturu **Init** vytvoříme instanci nového okna dané velikosti s daným umístěním na obrazovce.

Parametr **Rect** určuje umístění a velikost vytvářeného okna.

Parametr **S** označuje nadpis vytvářeného okna. Nadpis je umístěn uprostřed horního okraje okna. Pokud je nadpis širší, než je šířka okna, bude zaříznut.

Parametr **Frame** určuje jakým způsobem bude vytvářené okno orámováno. Jsou předdefinovány konstanty **FrameD** a **FrameS**. Pro orámování dvojitou čarou poslouží konstanta **FrameD**, pro orámování jednoduchou čarou konstanta **FrameS**.

Parametr **AS** je hodnota atributu barvy použitá při výpisu nadpisu S.

Parametr **AR** je hodnota atributu barvy použitá při orámování okna. Je-li AS = AR = 0 nebude vytvářené okno orámováno (pozn. Pro neorámované okno je ale efektivnější použít konstruktor **InitNoFrame** viz dále).

Parametr **AW** je hodnota atributu barvy použitá při výpisu textu v daném okně.

Význam atributů AS, AR a AW je popsán u proměnné **wA**.

Parametry **CL1** a **CL2** určují velikost (tvar) kurzoru použitého v daném okně. **CL1** určuje počáteční linku, **CL2** určuje koncovou linku kurzoru. Jsou-li hodnoty CL1 = \$20, CL2 = 0, nebude kurzor viditelný.

5.2.2. InitNoFrame

```
constructor InitNoFrame(Rect: TRect; AW: Byte; CL1, CL2: Byte);
```

Voláním konstrukturu **InitNoFrame** vytvoříme instanci nového okna dané velikosti s daným umístěním na obrazovce bez orámování. Parametry **Rect**, **AW**, **CL1** a **CL2** mají stejný význam jako u konstrukturu **Init** (viz výše).

5.2.3. Done

```
destructor Done;
```

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

5.2.4. 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.

5.2.5. 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).

5.2.6. PlaceFrame

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

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

5.2.7. SetCursor

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

Metoda pro nastavení velikosti (tvaru) kurzoru v daném okně. Lze ji volat i bez použití metod **LockWin** a **UnLockWin**. Parametry **CL1** a **CL2** určují velikost

kurzoru použitého v daném okně. **CL1** určuje počáteční linku, **CL2** určuje koncovou linku. Jsou-li hodnoty \$20, 0 nebude kurzor viditelný.

5.2.8. HiddenCursor

procedure HiddenCursor;

Voláním této metody je potlačen výpis kurzoru v daném okně. Toto volání je identické s voláním metody **SetCursor(\$20,\$0)**.

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, které k ní má vyhrazen přístup.

WaitKey : Word = 2;

WaitKey je počet wait stavů jádra Retos při čekání na stisk klávesy. Tato proměnná má význam pouze pro knihovnu **CrtWin**.

7. Popis procedur

7.1. CrtInit

procedure CrtInit

Procedura slouží k inicializaci knihovny **CrtWin** / **CrtWin2** a k vytvoření hlavní pracovní plochy o velikosti odpovídající maximální velikosti obrazovky aktuálního textového režimu. Na tuto plochu je možné se odkazovat prostřednictvím instance **MainWin**. Tuto proceduru je nutné zavolat před vytvořením jakékoliv instance objektu **tCrtWin**. V případě knihovny **CrtWin** se procedura **CrtInit** musí volat až po spuštění operačního systému Retos.

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.

Pozn: Tento příklad je k dispozici v elektronické formě v adresáři „EXAMPLES\CRTWIN\EXAMPLE1“.

```

program Example1_CrtWin;

uses
  Crt,
  Kernel,
  CrtWin;

const
  {počet procesu pro výpis na obrazovku}
  MaxProc1 = 3;
  {rozměry procesu pro výpis na obrazovku}
  Rect1 : array[1..MaxProc1] of TRect =
    ((X: 2; Y: 3; W: 38; H: 9),
     (X: 42; Y: 3; W: 38; H: 9),
     (X: 2; Y: 14; W: 38; H: 9));
  {rozměry procesu pro čtení klávesnice}
  Rect2 : TRect = (X: 42; Y: 14; W: 38; H: 9);

const
  FlEnd : Boolean = false; {příznak pro ukončení aplikace}

{$F+} { proces výpisu na obrazovku }
procedure P1(Rect: TRect; const S: String);
{$F-}
var
  Win : PCrtWin;
  i   : Word;
begin
  Start('Proc1',8000,90,254);Exit;
  { vytvoření okna }

Win:=New(PCrtWin,Init(Rect,S,FrameD,LightGray,LightGray,LightGray,$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;

{$F+} { proces vstupu z klávesnice }
procedure P2(Rect: TRect; const S: String);
{$F-}
var
  Win : PCrtWin;
  Key : char;
begin
  Start('Proc2',8000,90,254);Exit;
  { vytvoření okna }
Win:=New(PCrtWin,Init(Rect,S,FrameS,LightGray,LightGray,LightGray,{
$20,0}1,8));
  Win^.LockWin;   { získání práva přístupu na obrazovku }
  Write('Press any keys: '); { výpis na obrazovku }
  Win^.UnLockWin; { ukončení práva přístupu na obrazovku }

```

```
repeat
  Win^.LockWin;      { získání práva přístupu na obrazovku }
  if keypressed then { vstup z klávesnice }
  begin
    Key:=Readkey;
    if Key in [' '..#255] then write(Key);
    FlEnd:=Key=#$1B{ukončení na klávesu Esc};
  end;
  Wait(1);
  Win^.UnLockWin;   { ukončení práva přístupu na obrazovku }
until FlEnd;
end;

var
  i : Byte;
begin
  {nastavení videorežimu}
  TextMode(CO80);

  {inicializace o.s.Retos}
  StartMain(250,254);
  InitInterruptStack(1,8000);
  StartTimeSlicing($8);

  {inicializace výpisových oken}
  CrtInit;
  MainWin^.HiddenCursor; {skrytí kurzoru}
  MainWin^.LockWin;
  write('Example of CrtWin by SofCon');
  gotoxy(1,Hi(WindMax)+1); {jdi na poslední řádek}
  write('Help: Press <Esc> for Exit');
  MainWin^.UnLockWin;

  for i:=1 to MaxProcl do
    P1(Rect1[i],' Process 1 '); { spuštění procesu P1 }

    P2(Rect2,' Process 2 '); { spuštění procesu P2 }

  SetPriority(10,254); {nastavení priority hlavního procesu}

  repeat
    {tělo hlavního procesu}
  until FlEnd;

  {ukončení o.s.Retos}
  Abort('*');
  TerminateMain;
end.
```