

# LZ77

## KNIHOVNA PRO KOMPRESI A DEKOMPRESI DAT POMOCÍ ALGORITMU LZ77

Příručka uživatele a programátora



**SofCon<sup>®</sup> spol. s r.o.**  
Střešovická 49  
162 00 Praha 6  
tel/fax: +420 220 180 454  
E-mail: [sofcon@sofcon.cz](mailto: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: 17.06.2004

Datum posledního uložení dokumentu: 17.06.2004

(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	4
1.1.	Revize dokumentu	4
1.2.	Účel dokumentu	4
1.3.	Rozsah platnosti	4
1.4.	Související dokumenty	4
2.	Termíny a definice	4
3.	Úvod	5
4.	Reference	5
4.1.	Konstanty	5
4.1.1.	Konstanty lzs_	5
4.2.	Typy	5
4.2.1.	TLz77InputFunc	5
4.2.2.	Typ TLz77OutputFunc	6
4.2.3.	Struktura TLzBlockDescr	6
4.3.	Funkce	6
4.3.1.	Funkce Lz77CompressStm	6
4.3.2.	Funkce Lz77DecompressStm	7
4.3.3.	Funkce Lz77CompressBuff	8
4.3.4.	Funkce Lz77DecompressBuff	9
4.3.5.	Funkce Lz77DecompressBlocks	9
5.	Příklad	10

## 1. O dokumentu

---

### 1.1. Revize dokumentu

---

Verze dokumentu	Verze SW	Autor	Datum vydání	Popis změn
1.00	1.00	Cr	21.01.2004	První vydání
1.01	1.00	Wil, Bin	17.06.2004	Grafické úpravy dokumentu.

### 1.2. Účel dokumentu

---

Tento dokument slouží jako popis knihovny Lz77, která je součástí systémových knihoven firmy SofCon.

### 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 orientovat se 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

---

Knihovna LZ77 poskytuje funkce pro kompresi a dekompresi dat pomocí algoritmu LZ77. Funkce této knihovny lze použít pro kompresi nejrůznějších dat, jako jsou například bitmapy, texty apod. Vzhledem k použitému algoritmu komprese, se předpokládá, že komprese se bude provádět při vývoji aplikace na PC a v jednotce KIT se bude provádět už pouze dekomprese.

### 4. Reference

---

#### 4.1. Konstanty

---

##### 4.1.1. Konstanty lzs\_

Konstanty s prefixem lzs\_ slouží jako návratové kódy funkcí **Lz77Compress** a **Lz77Decompress** apod. (viz. kapitoly 4.3.1, 4.3.2);

<b>Identifikátor</b>	<b>Kód</b>	<b>Popis bitmapy</b>
lzsSuccess	0	Úspěšné provedení funkce.
lzsMemOvf	1	Nedostatek paměti na hromadě.
lzsInvPar	2	Neplatný parametr.
lzsError	3	Chyba dekomprese.
lzsEOF	4	Dosažen konec vstupních dat.
lzsInpError	5	Chyba vstupní operace.
lzsOutError	6	Chyba výstupní operace

#### 4.2. Typy

---

##### 4.2.1. TLz77InputFunc

Typ **TLz77InputFunc** je procedurální typ definující prototyp funkce předávané jako parametr funkcí **Lz77CompressStm** a **Lz77DecompressSts**. Tato funkce zajišťuje načtení jednoho znaku ze vstupu.

```
TLz77InputFunc = function ( var AData: Byte;
                             AContext: Pointer ): Integer;
```

##### Parametry:

AData	Do parametru AData musí funkce před návratem uložit jeden znak přečtený ze vstupu.
AContext	Uživatelský parametr, který je naplněn hodnotou parametru AInpContext funkce <b>Lz77CompressStm</b> nebo <b>Lz77DecompressStm</b> .

**Návratové hodnoty:**

V případě úspěšného načtení znaku ze vstupu musí funkce vrátit hodnotu `lzsSucess`. V případě dosažení konce vstupních dat musí vrátit hodnotu `lzsEof`. V případě, že dojde k nedostatku paměti při načítání znaku musí vrátit hodnotu `lzsMemOvf`. V případě jiné chyby vrací `lzsInpError`.

**4.2.2. Typ TLz77OutputFunc**

Typ **TLz77OutputFunc** je procedurální typ definující prototyp funkce předávané jako parametr funkcí **Lz77CompressStm** a **Lz77DecompressStm**. Tato funkce zajišťuje zápis jednoho znaku na výstup.

```
TLz77OutputFunc = function ( AData: Byte;
                             AContext: Pointer ): Integer;
```

**Parametry:**

AData	Parametr AData obsahuje hodnotu znaku zapisovaného na výstup.
AContext	Uživatelský parametr, který je naplněn hodnotou parametru AOutContext funkce <b>Lz77CompressStm</b> nebo <b>Lz77DecompressStm</b> .

**Návratové hodnoty:**

V případě úspěšného zapsání znaku na výstup musí funkce vrátit hodnotu `lzsSucess`. V případě, že dojde k nedostatku paměti při zápisu znaku, musí vrátit hodnotu `lzsMemOvf`. V případě jiné chyby vrací `lzsOutError`.

**4.2.3. Struktura TLzBlockDescr**

```
PLzBlockDescr = ^TLzBlockDescr;
TLzBlockDescr =
record
  Data : Pointer;
  Size : Word;
end;
```

**4.3. Funkce****4.3.1. Funkce Lz77CompressStm**

Funkce **Lz77CompressStm** provádí proudovou kompresi dat pomocí algoritmu LZ77.

```
function Lz77Compress(
  AInpFunc      : TLz77InputFunc;
  AInpContext   : Pointer;
  AOutFunc      : TLz77OutputFunc;
  AOutContext   : Pointer;
  AOfsBits     : Byte ): Integer;
```

**Parametry:**

AInpFunc	Vstupní funkce pro nekomprimovaná data. (viz. kapitola 4.2.1).
AInpContext	Uživatelský parametr předávaný vstupní funkci.
AOutFunc	Výstupní funkce pro komprimovaná data (viz. kapitola 4.2.2).
AOutContext	Uživatelský parametr předávaný výstupní funkci.
AOfsBits	Počet bitů na zakódování offsetu do plovoucího okna. Mocnina dvou tohoto parametru udává velikost okna ve znacích. Hodnota parametru se může pohybovat v rozsahu 9 až 14. Tato hodnota je součástí komprimovaných dat a při dekompresi již není potřeba tento parametr uvádět.

**Návratové hodnoty:**

V případě úspěšného provedení vrací funkce hodnotu `lzsSuccess`. V opačném případě vrací jednu z konstant:

<code>lzsInvPar</code>	V případě neplatného parametru <code>AOfsBits</code> .
<code>lzsMemOvf</code>	V případě nedostatku paměti na hromadě.
<code>lzsInpError</code>	V případě chyby v případě chyby vstupní funkce
<code>lzsOutError</code>	V případě chyby výstupní funkce.

**Poznámky:**

Funkce alokuje na hromadě prostor pro plovoucí okno, tj. 512B-16KB podle hodnoty parametru `AOfsBits`. Parametr `AOfsBits` ovlivňuje kvalitu komprese. Čím je tento parametr větší, tím větší je velikost plovoucího okna pro vyhledávání stejných sekvencí znaků (tj. lepší kompresní poměr). S rostoucím parametrem `AOfsBits` ovšem roste počet bitů pro zakódování offsetu do plovoucího okna. Pro různá data je tedy potřeba vyzkoušet různé hodnoty tohoto parametru. Optimum by mělo být někde kolem 11 až 12.

Vzhledem k použití této funkce, je algoritmus implementován triviálním způsobem, je použita hrubá síla a složitost algoritmu je  $O(n * 2^{2k})$  kde  $n = \text{AInSize}$  a  $k = \text{AOfsBits}$ .

### 4.3.2. Funkce `Lz77DecompressStm`

Funkce `Lz77DecompressStm` provádí proudovou dekompresi dat pomocí algoritmu LZ77.

```
function Lz77DecompressStm(
  AInpFunc      : TLz77InputFunc;
  AInpContext   : Pointer;
  AOutFunc      : TLz77OutputFunc;
  AOutContext   : Pointer ): Integer;
```

**Parametry:**

AInpFunc	Vstupní funkce pro komprimovaná data. (viz. kapitola 4.2.1).
AInpContext	Uživatelský parametr předávaný vstupní funkci.
AOutFunc	Výstupní funkce pro nekomprimovaná data (viz. kapitola 4.2.2).

AOutContext      Uživatelský parametr předávaný výstupní funkci.

### Návratové hodnoty:

V případě úspěšného provedení vrací funkce hodnotu lzSuccess. V opačném případě vrací jednu z konstant:

lzMemOvf      V případě nedostatku paměti na hromadě.  
 lzError        V případě chyby dekomprese způsobené porušenými daty.  
 lzInpError    V případě chyby v případě chyby vstupní funkce  
 lzOutError    V případě chyby výstupní funkce.

### Poznámky:

Funkce alokuje na hromadě prostor pro plovoucí okno, tj. 512B-16KB podle hodnoty parametru AOfsBits zadaného při kompresi.

Algoritmus má složitost  $O(n)$ , kde  $n$  je délka nekomprimovaných dat.

### 4.3.3. Funkce Lz77CompressBuff

Funkce **Lz77CompressBuff** provádí kompresi souvislého bloku dat pomocí algoritmu LZ77.

```
function Lz77CompressBuff(
  AInBuff   : Pointer;
  AInSize   : Word;
  AOutBuff  : Pointer;
  AOutSize  : Word;
  AOfsBits  : Byte ): Word;
```

### Parametry:

AInBuff        Ukazatel na blok nekomprimovaných dat.  
 AInSize        Délka nekomprimovaných dat.  
 AOutBuff      Ukazatel na buffer, do kterého budou uložena komprimovaná data.  
 AOutSize      Délka bufferu AOutBuff.  
 AOfsBits      Počet bitů na zakódování offset do plovoucího okna. Mocnina dvou tohoto parametru udává velikost okna ve znacích. Hodnota parametru se může pohybovat v rozsahu 9 až 14.

### Návratové hodnoty:

V případě úspěšného provedení vrací funkce počet bajtů uložených do výstupního bufferu. V případě chyby, např. nedostatku paměti nebo nedostatečně velkého výstupního bufferu funkce vrací hodnotu 65535 (\$FFFF).

### Poznámky:

V současné implementaci knihovny funkce **Lz77CompressBuff** interně volá funkci **Lz77Compress**.

Funkce alokuje na hromadě prostor pro plovoucí okno, tj. 512B-16KB podle hodnoty parametru AOfsBits. Parametr AOfsBits ovlivňuje kvalitu komprese. Čím je tento parametr větší, tím větší je velikost plovoucího okna pro vyhledávání stejných sekvencí znaků (tj. lepší kompresní poměr). S rostoucím parametrem AOfsBits ovšem



roste počet bitů pro zakódování offsetu do plovoucího okna. Pro různá data je tedy potřeba vyzkoušet různé hodnoty tohoto parametru. Optimum by mělo být někde kolem 11 až 12.

Vzhledem k použití této funkce, je algoritmus implementován triviálním způsobem, je použita hrubá síla a složitost algoritmu je  $O(n * 2^{2k})$  kde  $n = AInSize$  a  $k = AOfsBits$ .

#### 4.3.4. Funkce Lz77DecompressBuff

Funkce **Lz77CompressBuff** provádí dekompresi souvislého bloku dat pomocí algoritmu LZ77.

```
function Lz77DecompressBuff(
  AInBuff   : Pointer;
  AInSize   : Word;
  AOutBuff  : Pointer;
  AOutSize  : Word ): Word;
```

##### Parametry:

AInBuff	Ukazatel na blok komprimovaných dat.
AInSize	Délka komprimovaných dat. V případě, že je tento parametr nastaven na hodnotu větší, než je délka vstupních dat, např. 65535, bude dekomprese správně ukončena nalezením koncové značky dat.
AOutBuff	Ukazatel na buffer, do kterého budou uložena dekomprimovaná data.
AOutSize	Délka bufferu AOutBuff.

##### Návratové hodnoty:

V případě úspěšného provedení vrací funkce počet bajtů uložených do výstupního bufferu. V případě chyby, např. nedostatečně velkého výstupního bufferu, nebo v případě porušených vstupních dat funkce vrací hodnotu 65535 (\$FFFF).

##### Poznámky:

Funkce **Lz77DecompressBuff** je napsána jako velice rychlá funkce nevyžadující žádnou dodatečnou paměť na hromadě (na rozdíl od funkce **Lz77Decompress**). Funkce je však omezena pouze na 64kB jak komprimovaných tak dekomprimovaných dat.

#### 4.3.5. Funkce Lz77DecompressBlocks

Funkce **Lz77CompressBlocks** provádí dekompresi seznamu bloků dat pomocí algoritmu LZ77.

```
function Lz77DecompressBlocks( AInData: PLzBlockDescr;
  AOutFunc: TLz77OutputFunc; AOutContext: Pointer ): Integer;
```

**Parametry:**

AInData	Ukazatel na seznam bloků komprimovaných dat.
AOutFunc	Výstupní funkce pro nekomprimovaná data (viz. kapitola 4.2.2).
AOutContext	Uživatelský parametr předávaný výstupní funkci.

**Návratové hodnoty:**

V případě úspěšného provedení vrací funkce hodnotu lzSuccess. V opačném případě vrací jednu z konstant:

lzMemOvf	V případě nedostatku paměti na hromadě.
lzError	V případě chyby dekomprese způsobené porušenými daty.
lzInpError	V případě chyby v případě chyby vstupní funkce
lzOutError	V případě chyby výstupní funkce.

**Poznámky:**

Funkce alokuje na hromadě prostor pro plovoucí okno, tj. 512B-16KB podle hodnoty parametru AOfsBits zadaného při kompresi.

Algoritmus má složitost  $O(n)$ , kde  $n$  je délka nekomprimovaných dat.

## 5. Příklad

---

Použití funkcí Lz77Compress a Lz77Decompress se může zdát na první pohled komplikované, ale není tomu tak, jak ukazuje následující příklad, který provádí kompresi a dekompresi souvislého bloku dat pomocí těchto dvou funkcí.

```

type
  PInOutContext = ^TInOutContext;
  TInOutContext = record
    Buff      : PChar;
    Size      : Word;
    Offset    : Word;
  end;

function InpFunc( var AData: Byte; AContext: Pointer ): Integer;
  far;
begin
  with PInOutContext( AContext )^ do
  begin
    if Offset < Size then
    begin
      AData := Ord( (Buff + Offset)^ );
      Inc( Offset );
      InpFunc := lzSuccess;
    end
    else begin
      InpFunc := lzEof;
    end;
  end;
end;

function OutFunc( AData: Byte; AContext: Pointer ): Integer; far;
begin
  with PInOutContext( AContext )^ do
  begin

```

```
    if Offset < Size then
    begin
        (Buff + Offset )^ := Char( AData );
        Inc( Offset );
        OutFunc := lzsSuccess;
    end
    else begin
        OutFunc := lzsOutError;
    end;
end;
end;
end;

var
    InpCtx   : TInOutContext;
    OutCtx   : TInOutContext;

    InBuff   : array[0..100] of Byte;
    OutBuff  : array[0..100] of Byte;

    Status   : Integer;

begin
    { Komprese InBuff -> OutBuff }

    InpCtx.Buff   := @InBuff;
    InpCtx.Size   := SizeOf( InBuff );
    InpCtx.Offset := 0;

    OutCtx.Buff   := @OutBuff;
    OutCtx.Size   := SizeOf( OutBuff );
    OutCtx.Offset := 0;

    Status := Lz77Compress( InpFunc, @InpCtx, OutFunc, @OutCtx, 11 );

    case Status of
        lzsSuccess :
            WriteLn( 'OK, delka kompr. dat = ', OutCtx.Offset );
    else
        WriteLn('Error' );
        Exit;
    end;

    { Dekomprese OutBuff -> InBuff }

    InpCtx.Buff   := @OutBuff;
    InpCtx.Size   := OutCtx.Offset;
    InpCtx.Offset := 0;

    OutCtx.Buff   := @InBuff;
    OutCtx.Size   := SizeOf( InBuff );
    OutCtx.Offset := 0;

    Status := Lz77Decompress( InpFunc, @InpCtx, OutFunc, @OutCtx );

    case Status of
        lzsSuccess :
            WriteLn( 'OK, delka dekompr. dat = ', OutCtx.Offset );
    else
        WriteLn('Error' );
        Exit;
    end;

end.
```