Function DevPosSensitive: Compilation with and without Loop Alignment

(Abstract: Loop alignment makes the function 12 % faster, all alignments are placed to be **not** runtime consuming by themselves)

```
function DevPosSensitive(const SubStr, S : ShortString) : SmallInt;
var
  CntS,
  CntSub,
  LenS,
  LenSub : Byte;
  PS,
  PSTmp,
  PSub   : PByte;
begin
  result := 0;
  LenSub := Length(SubStr);
  LenS   := Length(S);
  if (LenS < LenSub) or (LenSub = 0) then exit;
  dec(LenS, LenSub - 1);
  CntS := LenS;
  PS := @S[1];
  PSub := @SubStr[1];
  Case LenSub of
      1 : while PSub^ <> PS^ do
          begin
            dec(CntS);
            if CntS = 0 then exit;
            inc(PS);
          end;
      2 : while PWord(PSub)^ <> PWord(PS)^ do
          begin
            dec(CntS);
            if CntS = 0 then exit;
            inc(PS);
          end;
      3 : while (PWord(PSub)^ <> PWord(PS)^) or (PByte((PSub+2)^) <> PByte((PS+2)^)) do
          begin
            dec(CntS);
            if CntS = 0 then exit;
            inc(PS);
          end;
      4 : while PLongWord(PSub)^ <> PLongWord(PS)^ do
          begin
            dec(CntS);
            if CntS = 0 then exit;
            inc(PS);
          end;
```

```
    else Repeat
           while PLongWord(PSub)^ <> PLongWord(PS)^ do
           begin
             dec(CntS);
             if CntS = 0 then exit;
             inc(PS);
           end;
           PSTmp := PS + 4;
           inc(PSub, 4);
           CntSub := LenSub - 4;
           while PSTmp^ = PSub^ do
           begin
             dec(CntSub);
             if CntSub = 0 then begin result := LenS - CntS + 1; exit; end;
             inc(PSTmp);
             inc(PSub);
           end;
           dec(CntS);
           if CntS = 0 then exit;
           inc(PS);
           PSub := @SubStr[1];
         until true = false;
  end;
  result := LenS - CntS + 1;
end;
```

{$CODEALIGN PROC=32}
{$CODEALIGN LOOP=16}

Addr	Opcode	Instr	Param	Target
begin				
CD40	4883ec18	sub	$0x18,%rsp	
CD44	48891c24	mov	%rbx,(%rsp)	
CD48	4c89642408	mov	%r12,0x8(%rsp)	
result := 0;				
CD4D	66b80000	mov	$0x0,%ax	
LenSub := Length(SubStr);				
CD51	448a07	mov	(%rdi),%r8b	
LenS := Length(S);				
CD54	8a0e	mov	(%rsi),%cl	
if (LenS < LenSub) or (LenSub = 0) then exit;				

{$CODEALIGN PROC=32}
No Codealign Loop

Addr	Opcode	Instr	Param	Target
CD40	4883ec18	sub	$0x18,%rsp	
CD44	48891c24	mov	%rbx,(%rsp)	
CD48	4c89642408	mov	%r12,0x8(%rsp)	
CD4D	66b80000	mov	$0x0,%ax	
CD51	448a07	mov	(%rdi),%r8b	
CD54	8a0e	mov	(%rsi),%cl	

```
CD56    88ca            mov     %cl,%dl
CD58    4438c2          cmp     %r8b,%dl
CD5B    0f82e2010000    jb      0x83cf43        <+515>
CD61    410fb6d0        movzbl  %r8b,%edx
CD65    4885d2          test    %rdx,%rdx
CD68    0f84d5010000    je      0x83cf43        <+515>
```

```
  dec(LenS, LenSub - 1);
```

```
CD6E    410fb6d0        movzbl  %r8b,%edx
CD72    48ffca          dec     %rdx
CD75    28d1            sub     %dl,%cl
```

```
  CntS := LenS;
```

```
CD77    88ca            mov     %cl,%dl
```

```
  PS := @S[1];
```

```
CD79    488d7601        lea     0x1(%rsi),%rsi
```

```
  PSub := @SubStr[1];
```

```
CD7D    4c8d4f01        lea     0x1(%rdi),%r9
```

```
  Case LenSub of
```

```
CD81    4588c2          mov     %r8b,%r10b
CD84    4180fa01        cmp     $0x1,%r10b
CD88    0f8234010000    jb      0x83cec2        <+386>
CD8E    4588d3          mov     %r10b,%r11b
CD91    41feca          dec     %r10b
CD94    4180fb01        cmp     $0x1,%r11b
CD98    7458            je      0x83cdf2        <+178>
CD9A    4588d3          mov     %r10b,%r11b
CD9D    41feca          dec     %r10b
CDA0    4180fb01        cmp     $0x1,%r11b
CDA4    0f8478000000    je      0x83ce22        <+226>
CDAA    4588d3          mov     %r10b,%r11b
CDAD    41feca          dec     %r10b
CDB0    4180fb01        cmp     $0x1,%r11b
CDB4    0f8498000000    je      0x83ce52        <+274>
CDBA    4588d3          mov     %r10b,%r11b
CDBD    41feca          dec     %r10b
CDC0    4180fb01        cmp     $0x1,%r11b
CDC4    0f84c8000000    je      0x83ce92        <+338>
CDCA    e9f3000000      jmpq    0x83cec2        <+386>
CDCF    eb21            jmp     0x83cdf2        <+178>
CDD1    66666690        data32 data32 xchg %ax,%ax
CDD5    66666690        data32 data32 xchg %ax,%ax
CDD9    66666690        data32 data32 xchg %ax,%ax
CDDD    666690          data32 xchg %ax,%ax
```

15 Byte NOP

```
CD56    88ca            mov     %cl,%dl
CD58    4438c2          cmp     %r8b,%dl
CD5B    0f82a6010000    jb      0x83cf07        <+455>
CD61    410fb6d0        movzbl  %r8b,%edx
CD65    4885d2          test    %rdx,%rdx
CD68    0f8499010000    je      0x83cf07        <+455>
```

```
CD6E    410fb6d0        movzbl  %r8b,%edx
CD72    48ffca          dec     %rdx
CD75    28d1            sub     %dl,%cl
```

```
CD77    88ca            mov     %cl,%dl
```

```
CD79    488d7601        lea     0x1(%rsi),%rsi
```

```
CD7D    4c8d4f01        lea     0x1(%rdi),%r9
```

```
CD81    4588c2          mov     %r8b,%r10b
CD84    4180fa01        cmp     $0x1,%r10b
CD88    0f8200010000    jb      0x83ce8e        <+334>
CD8E    4588d3          mov     %r10b,%r11b
CD91    41feca          dec     %r10b
CD94    4180fb01        cmp     $0x1,%r11b
CD98    7448            je      0x83cde2        <+162>
CD9A    4588d3          mov     %r10b,%r11b
CD9D    41feca          dec     %r10b
CDA0    4180fb01        cmp     $0x1,%r11b
CDA4    7460            je      0x83ce06        <+198>
CDA6    4588d3          mov     %r10b,%r11b
CDA9    41feca          dec     %r10b
CDAC    4180fb01        cmp     $0x1,%r11b
CDB0    0f8474000000    je      0x83ce2a        <+234>
CDB6    4588d3          mov     %r10b,%r11b
CDB9    41feca          dec     %r10b
CDBC    4180fb01        cmp     $0x1,%r11b
CDC0    0f84a4000000    je      0x83ce6a        <+298>
CDC6    e9c3000000      jmpq    0x83ce8e        <+334>
CDCB    eb15            jmp     0x83cde2        <+162>
CDCD    666690          data32 xchg %ax,%ax
```

3 Byte NOP

```
      1 : while PSub^ <> PS^ do
          begin
            dec(CntS);
```

```
CDE0    feca            dec    %dl
```

```
            if CntS = 0 then exit;
```

```
CDE2    440fb6d2        movzbl %dl,%r10d
CDE6    4d85d2          test   %r10,%r10
CDE9    0f8454010000    je     0x83cf43        <+515>
```

```
            inc(PS);
          end;
```

```
CDEF    48ffc6          inc    %rsi
CDF2    458a11          mov    (%r9),%r10b
CDF5    443a16          cmp    (%rsi),%r10b
CDF8    75e6            jne    0x83cde0        <+160>
CDFA    e935010000      jmpq   0x83cf34        <+500>
CDFF    eb21            jmp    0x83ce22        <+226>
CE01    66666690        data32 data32 xchg %ax,%ax
CE05    66666690        data32 data32 xchg %ax,%ax
CE09    66666690        data32 data32 xchg %ax,%ax
CE0D    666690          data32 xchg %ax,%ax
```

15 Byte NOP

```
      2 : while PWord(PSub)^ <> PWord(PS)^ do
          begin
            dec(CntS);
            if CntS = 0 then exit;
            inc(PS);
          end;
```

```
CE10    feca            dec    %dl
CE12    440fb6d2        movzbl %dl,%r10d
CE16    4d85d2          test   %r10,%r10
CE19    0f8424010000    je     0x83cf43        <+515>
CE1F    48ffc6          inc    %rsi
CE22    66458b11        mov    (%r9),%r10w
CE26    66443b16        cmp    (%rsi),%r10w
CE2A    75e4            jne    0x83ce10        <+208>
CE2C    e903010000      jmpq   0x83cf34        <+500>
CE31    eb1f            jmp    0x83ce52        <+274>
CE33    66666690        data32 data32 xchg %ax,%ax
CE37    66666690        data32 data32 xchg %ax,%ax
CE3B    66666690        data32 data32 xchg %ax,%ax
CE3F    90              nop
```

13 Byte NOP

```
CDD0    feca            dec    %dl
```

```
CDD2    440fb6d2        movzbl %dl,%r10d
CDD6    4d85d2          test   %r10,%r10
CDD9    0f8428010000    je     0x83cf07        <+455>
```

```
CDDF    48ffc6          inc    %rsi
CDE2    458a11          mov    (%r9),%r10b
CDE5    443a16          cmp    (%rsi),%r10b
CDE8    75e6            jne    0x83cdd0        <+144>
CDEA    e909010000      jmpq   0x83cef8        <+440>
CDEF    eb15            jmp    0x83ce06        <+198>
CDF1    666690          data32 xchg %ax,%ax
```

3 Byte NOP

```
CDF4    feca            dec    %dl
CDF6    440fb6d2        movzbl %dl,%r10d
CDFA    4d85d2          test   %r10,%r10
CDFD    0f8404010000    je     0x83cf07        <+455>
CE03    48ffc6          inc    %rsi
CE06    66458b11        mov    (%r9),%r10w
CE0A    66443b16        cmp    (%rsi),%r10w
CE0E    75e4            jne    0x83cdf4        <+180>
CE10    e9e3000000      jmpq   0x83cef8        <+440>
CE15    eb13            jmp    0x83ce2a        <+234>
CE17    90              nop
```

1 Byte NOP

```
      3 : while (PWord(PSub)^ <> PWord(PS)^) or (PByte((PSub+2)^) <> PByte((PS+2)^)) do
          begin
            dec(CntS);
            if CntS = 0 then exit;
            inc(PS);
          end;
```

```
CE40    feca            dec    %dl
CE42    440fb6d2        movzbl %dl,%r10d
CE46    4d85d2          test   %r10,%r10
CE49    0f84f4000000    je     0x83cf43        <+515>
CE4F    48ffc6          inc    %rsi
CE52    66458b11        mov    (%r9),%r10w
CE56    66443b16        cmp    (%rsi),%r10w
CE5A    75e4            jne    0x83ce40        <+256>
CE5C    4d89ca          mov    %r9,%r10
CE5F    4983c202        add    $0x2,%r10
CE63    450fb61a        movzbl (%r10),%r11d
CE67    4989f2          mov    %rsi,%r10
CE6A    4983c202        add    $0x2,%r10
CE6E    450fb612        movzbl (%r10),%r10d
CE72    4d39d3          cmp    %r10,%r11
CE75    75c9            jne    0x83ce40        <+256>
CE77    e9b8000000      jmpq   0x83cf34        <+500>
CE7C    eb14            jmp    0x83ce92        <+338>
CE7E    6690            xchg   %ax,%ax                 2 Byte NOP
```

```
CE18    feca            dec    %dl
CE1A    440fb6d2        movzbl %dl,%r10d
CE1E    4d85d2          test   %r10,%r10
CE21    0f84e0000000    je     0x83cf07        <+455>
CE27    48ffc6          inc    %rsi
CE2A    66458b11        mov    (%r9),%r10w
CE2E    66443b16        cmp    (%rsi),%r10w
CE32    75e4            jne    0x83ce18        <+216>
CE34    4d89ca          mov    %r9,%r10
CE37    4983c202        add    $0x2,%r10
CE3B    450fb61a        movzbl (%r10),%r11d
CE3F    4989f2          mov    %rsi,%r10
CE42    4983c202        add    $0x2,%r10
CE46    450fb612        movzbl (%r10),%r10d
CE4A    4d39d3          cmp    %r10,%r11
CE4D    75c9            jne    0x83ce18        <+216>
CE4F    e9a4000000      jmpq   0x83cef8        <+440>
CE54    eb14            jmp    0x83ce6a        <+298>
CE56    6690            xchg   %ax,%ax                 2 Byte NOP
```

```
      4 : while PLongWord(PSub)^ <> PLongWord(PS)^ do
          begin
            dec(CntS);
            if CntS = 0 then exit;
            inc(PS);
          end;
```

```
CE80    feca            dec    %dl
CE82    440fb6d2        movzbl %dl,%r10d
CE86    4d85d2          test   %r10,%r10
CE89    0f84b4000000    je     0x83cf43        <+515>
CE8F    48ffc6          inc    %rsi
CE92    458b11          mov    (%r9),%r10d
CE95    443b16          cmp    (%rsi),%r10d
CE98    75e6            jne    0x83ce80        <+320>
CE9A    e995000000      jmpq   0x83cf34        <+500>
CE9F    90              nop
CEA0    eb20            jmp    0x83cec2        <+386>
CEA2    66666690        data32 data32 xchg %ax,%ax
CEA6    66666690        data32 data32 xchg %ax,%ax
CEAA    66666690        data32 data32 xchg %ax,%ax
CEAE    6690            xchg   %ax,%ax                 14 Byte NOP
```

```
CE58    feca            dec    %dl
CE5A    440fb6d2        movzbl %dl,%r10d
CE5E    4d85d2          test   %r10,%r10
CE61    0f84a0000000    je     0x83cf07        <+455>
CE67    48ffc6          inc    %rsi
CE6A    458b11          mov    (%r9),%r10d
CE6D    443b16          cmp    (%rsi),%r10d
CE70    75e6            jne    0x83ce58        <+280>
CE72    e981000000      jmpq   0x83cef8        <+440>
CE77    90              nop
CE78    eb14            jmp    0x83ce8e        <+334>
CE7A    66              xchg   %ax,%ax                 1 Byte NOP
```

```
    else Repeat
           while PLongWord(PSub)^ <> PLongWord(PS)^ do
           begin
             dec(CntS);
```

```
CEB0    feca            dec    %dl
```

```
CE7C    feca            dec    %dl
```

```
             if CntS = 0 then exit;
```

```
CEB2    440fb6d2        movzbl %dl,%r10d
CEB6    4d85d2          test   %r10,%r10
CEB9    0f8484000000    je     0x83cf43        <+515>
```

```
CE7E    440fb6d2        movzbl %dl,%r10d
CE82    4d85d2          test   %r10,%r10
CE85    0f847c000000    je     0x83cf07        <+455>
```

```
             inc(PS);
           end;
```

```
CEBF    48ffc6          inc    %rsi
CEC2    458b11          mov    (%r9),%r10d
CEC5    443b16          cmp    (%rsi),%r10d
CEC8    75e6            jne    0x83ceb0        <+368>
```

```
CE8B    48ffc6          inc    %rsi
CE8E    458b11          mov    (%r9),%r10d
CE91    443b16          cmp    (%rsi),%r10d
CE94    75e6            jne    0x83ce7c        <+316>
```

```
           PSTmp := PS + 4;
```

```
CECA    4989f2          mov    %rsi,%r10
CECD    4983c204        add    $0x4,%r10
CED1    4d89d3          mov    %r10,%r11
```

```
CE96    4989f2          mov    %rsi,%r10
CE99    4983c204        add    $0x4,%r10
CE9D    4d89d3          mov    %r10,%r11
```

```
           inc(PSub, 4);
```

```
CED4    4983c104        add    $0x4,%r9
```

```
CEA0    4983c104        add    $0x4,%r9
```

```
           CntSub := LenSub - 4;
```

```
CED8    450fb6d0        movzbl %r8b,%r10d
CEDC    4983ea04        sub    $0x4,%r10
CEE0    4488d3          mov    %r10b,%bl
CEE3    eb30            jmp    0x83cf15        <+469>
CEE5    66666690        data32 data32 xchg %ax,%ax
CEE9    66666690        data32 data32 xchg %ax,%ax      11 Byte NOP
CEED    666690          data32 xchg %ax,%ax
```

```
CEA4    450fb6d0        movzbl %r8b,%r10d
CEA8    4983ea04        sub    $0x4,%r10
CEAC    4488d3          mov    %r10b,%bl
CEAF    eb28            jmp    0x83ced9        <+409>
CEB1    666690          data32 xchg %ax,%ax             3 Byte NOP
```

```
           while PSTmp^ = PSub^ do
           begin
             dec(CntSub);
```

```
CEF0    fecb            dec    %bl
```

```
CEB4    fecb            dec    %bl
```

```
     if CntSub = 0 then begin result := LenS - CntS + 1; exit; end;
```

```
CEF2    440fb6d3        movzbl %bl,%r10d
CEF6    4d85d2          test   %r10,%r10
CEF9    7514            jne    0x83cf0f        <+463>
CEFB    440fb6d1        movzbl %cl,%r10d
CEFF    440fb6e2        movzbl %dl,%r12d
CF03    4d29e2          sub    %r12,%r10
```

```
CEB6    440fb6d3        movzbl %bl,%r10d
CEBA    4d85d2          test   %r10,%r10
CEBD    7514            jne    0x83ced3        <+403>
CEBF    440fb6d1        movzbl %cl,%r10d
CEC3    440fb6e2        movzbl %dl,%r12d
CEC7    4d29e2          sub    %r12,%r10
```

```
CF06    49ffc2          inc    %r10                                  CECA    49ffc2          inc    %r10
CF09    664489d0        mov    %r10w,%ax                             CECD    664489d0        mov    %r10w,%ax
CF0D    eb34            jmp    0x83cf43        <+515>                CED1    eb34            jmp    0x83cf07        <+455>
      inc(PSTmp);
CF0F    49ffc3          inc    %r11                                  CED3    49ffc3          inc    %r11
      inc(PSub);
CF12    49ffc1          inc    %r9                                   CED6    49ffc1          inc    %r9
          end;
CF15    458a13          mov    (%r11),%r10b                          CED9    458a13          mov    (%r11),%r10b
CF18    453a11          cmp    (%r9),%r10b                           CEDC    453a11          cmp    (%r9),%r10b
CF1B    74d3            je     0x83cef0        <+432>                CEDF    74d3            je     0x83ceb4        <+372>
          dec(CntS);
CF1D    feca            dec    %dl                                   CEE1    feca            dec    %dl
          if CntS = 0 then exit;
CF1F    440fb6d2        movzbl %dl,%r10d                             CEE3    440fb6d2        movzbl %dl,%r10d
CF23    4d85d2          test   %r10,%r10                             CEE7    4d85d2          test   %r10,%r10
CF26    741b            je     0x83cf43        <+515>                CEEA    741b            je     0x83cf07        <+455>
          inc(PS);
CF28 48ffc6          inc    %rsi                                     CEEC    48ffc6          inc    %rsi
          PSub := @SubStr[1];
CF2B    4c8d5701        lea    0x1(%rdi),%r10                        CEEF    4c8d5701        lea    0x1(%rdi),%r10
CF2F    4d89d1          mov    %r10,%r9                              CEF3    4d89d1          mov    %r10,%r9
        until true = false;
  end;
CF32    eb8e            jmp    0x83cec2        <+386>                CEF6    eb96            jmp    0x83ce8e        <+334>
  result := LenS - CntS + 1;
CF34    0fb6c9          movzbl %cl,%ecx                              CEF8    0fb6c9          movzbl %cl,%ecx
CF37    0fb6d2          movzbl %dl,%edx                              CEFB    0fb6d2          movzbl %dl,%edx
CF3A    4829d1          sub    %rdx,%rcx                             CEFE    4829d1          sub    %rdx,%rcx
CF3D    48ffc1          inc    %rcx                                  CF01    48ffc1          inc    %rcx
CF40    6689c8          mov    %cx,%ax                               CF04    6689c8          mov    %cx,%ax
end;
CF43    0fbfc0          movswl %ax,%eax                              CF07    0fbfc0          movswl %ax,%eax
CF46    488b1c24        mov    (%rsp),%rbx                           CF0A    488b1c24        mov    (%rsp),%rbx
CF4A    4c8b642408      mov    0x8(%rsp),%r12                        CF0E    4c8b642408      mov    0x8(%rsp),%r12
CF4F    4883c418        add    $0x18,%rsp                            CF13    4883c418        add    $0x18,%rsp
CF53    c3666666906666690666666 retq                                 CF17    c36666669066666690       retq
```

```
CF54    66666690          data32 data32 xchg %ax,%ax
CF58    66666690          data32 data32 xchg %ax,%ax
CF5C    66666690          data32 data32 xchg %ax,%ax
```
12 Byte NOP

```
CF18    66666690          data32 data32 xchg %ax,%ax
CF1C    66666690          data32 data32 xchg %ax,%ax
```
8 Byte NOP

Test results:

Case-Sensitive substring search using several Pos functions:

Number of Strings:	1 025 044
Number of Substrings:	17
Number of Runs:	1
Number of Pos calls:	17 425 748
Errors compared to System.Pos:	0
FPC Optimization Level:	2

String Dataset: All not-empty & not-duplicated lines of all *.pas; *.pp; *.p; *.inc; *.lpr Files in /usr/share/lazarus/1.2.4/
- 3 768 Files
- 1 025 044 Strings
- Leading and trailing tabs and spaces removed

Searchstrings used: 'L', 'Li', 'Lis', 'List', 'gList', 'ngList', 'ingList', 'ringList', 'tringList', 'StringList', 'sTRINGlIST', 'Length', 'lEnGt', 'eaccessviolation', 'AccessViolation', '', 'ccessviolatioN', '', 'Exit'

Function used	Time	Time/Call	Calls	Factor
System.Pos (by Call with Shortstrings):	2.886 s	166 ns	17425748	1.00
System.Pos (Cloned Shortstring Pas-Version):	2.816 s	162 ns	17425748	0.98
DevPosSensitive (With Loop Alignment):	**1.899 s**	**109 ns**	**17425748**	**0.66**
DevPosSensitive (Without Loop Alignment):	**2.148 s**	**123 ns**	**17425748**	**0.74**

Tested with: AMD 64
Debian Wheezy
Lazarus: 1.2.4
FPC: 2.6.4
SVN-Rev.: 45510
x86_64-linux-gtk 2

2015-02-02
Rüdiger Walter
(rue.walter@web.de)