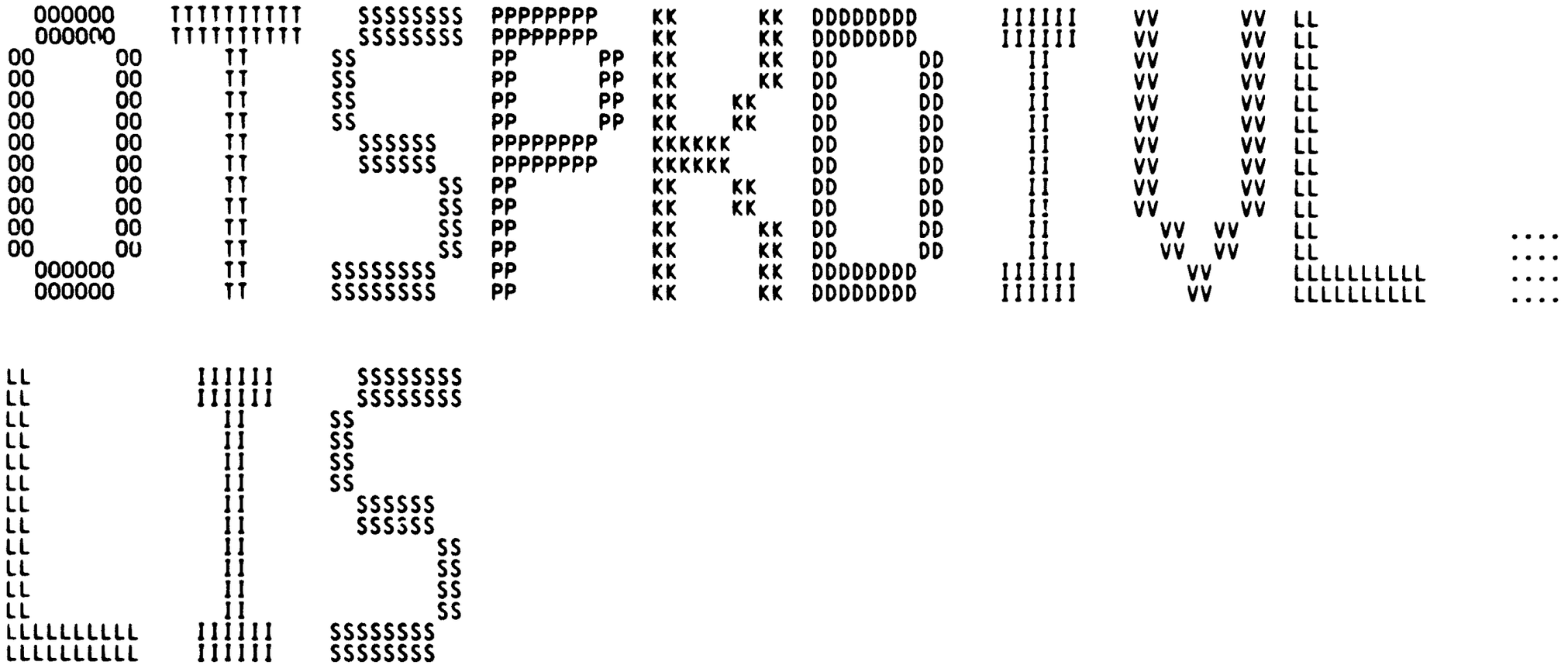
```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

FILEID**OTSPKDIVL

0 1



```
0000     1          .title  ots$div_pk_long
0000     2          .ident  /1-001/                                 ; Edit DG1001
0000     3 ;
0000     4 ;*****************************************************************************
0000     5 ;*                                                                           *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8 ;*  ALL RIGHTS RESERVED.                                                     *
0000     9 ;*                                                                           *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    15 ;*  TRANSFERRED.                                                             *
0000    16 ;*                                                                           *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    19 ;*  CORPORATION.                                                             *
0000    20 ;*                                                                           *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23 ;*                                                                           *
0000    24 ;*                                                                           *
0000    25 ;*****************************************************************************
0000    26 ;
0000    27 ;++
0000    28 ;
0000    29 ;
0000    30 ; routine:
0000    31 ;
0000    32 ;       OTS$DIV_PK_LONG
0000    33 ;
0000    34 ;
0000    35 ; facility:
0000    36 ;
0000    37 ;       VAX/VMS OTS runtime library.
0000    38 ;
0000    39 ; abstract:
0000    40 ;
0000    41 ;       Runtime routine performs fixed decimal (packed decimal) division.
0000    42 ;       The routine is called  when precision and scale  requirements for
0000    43 ;       the quotient imply multiple  precision  division.  The routine is
0000    44 ;       only called when such multiple precision division is required and
0000    45 ;       when the divisor has a precision of  30 or 31 decimal digits.
0000    46 ;       (Call   ots$div_pkshort   if   multiple   precision  division  is
0000    47 ;       required and the divisor has precision less than 30 decimal digits).
0000    48 ;
0000    49 ; author: Peter Baum  30-jun-1980
0000    50 ;
0000    51 ; modifications:
0000    52 ;
0000    53 ;
0000    54 ;       1-001   Debess Grabazs  5-March-1984
0000    55 ;               Change PLI routine to OTS routine.
0000    56 ;
0000    57 ;
```

```
0000     58 ;
0000     59 ;
0000     60 ;
0000     61 ; documentation file: THEORY.MEM
0000     62 ;
0000     63 ; functional description:
0000     64 ;
0000     65 ;       This routine calculates:
0000     66 ;
0000     67 ;       z = x / y
0000     68 ;
0000     69 ;       let a = scale(z) + scale(y) - scale(x)  - 31 + prec(x)
0000     70 ;           b = scale(z) + scale(y) - scale(x) + prec(x)
0000     71 ;           c = 31 - prec(x)
0000     72 ;           d = 31 - prec(y)
0000     73 ;
0000     74 ;       this routine is called if b > 31 and d < 2
0000     75 ;
0000     76 ;       Prior to the call:
0000     77 ;               if c not 0 then shift x left by c.
0000     78 ;               Thus x is a 31 digit packed decimal.
0000     79 ;
0000     80 ;
0000     81 ;
0000     82 ;
0000     83 ;       input:
0000     84 ;               0(ap)   # of arguments
0000     85 ;               4(ap)   address of dividend (shifted left by c)
0000     86 ;               8(ap)   address of divisor
0000     87 ;               12(ap)  precision of divisor (high order bytes zeroed)
0000     88 ;               16(ap)  address of quotient
0000     89 ;               20(ap)  precision of quotient (high order bytes zeroed)
0000     90 ;               24(ap)  a as defined above(high order bytes zeroed)
0000     91 ;
0000     92 ;
0000     93 ;       output:
0000     94 ;               quotient returned at address specified by 16(ap)
0000     95 ;
0000     96 ;
0000     97 ; optimization notes:
0000     98 ;
0000     99 ;       1) Optimized for speed, not space.
0000    100 ;       2) Optimized for y > 0.
0000    101 ;       3) Assumes speed for register to register operations are the same
0000    102 ;          for byte operations and longword operations.
0000    103 ;       4) Many  packed  instruction  sequences were timed. Do not change
0000    104 ;          unless  actual  tests  are  made  to determine relative speed.
0000    105 ;          Tests were made on 11/780 and Comet.
0000    106 ;
0000    107 ;
0000    108 ;
0000    109 ; possible optimizations:
0000    110 ;
0000    111 ;       1) currently  we  always calculate the next 15 digits each
0000    112 ;       iteration  and then truncate the last iteration as part of
0000    113 ;       the  final  step.   We  might be able to go through making
0000    114 ;       calculations with fewer digits on this last pass.
```

```
0000   115 ;
0000   116 ;       2) the classical trade-off involving flow paths which
0000   117 ;       could be merged (sometimes resulting in things like
0000   118 ;       adding 0) or merged with switches or left alone (resulting
0000   119 ;       in space-speed tradeoff in favor of space). Since we are
0000   120 ;       running on a virtual machine, even the speed factor is not
0000   121 ;       obvious.  This  came  up when remainder was 0, when borrow
0000   122 ;       was  not  needed,  when less than 15 digits where required
0000   123 ;       for  the  last  part  of  the  quotient,  and when the new
0000   124 ;       algorithm  had  found  precisely  what  the next 15 digits
0000   125 ;       were  but  did not have a value yet for the remainder (the
0000   126 ;       other algorithms always have a remainder when the quotient
0000   127 ;       is known).
0000   128 ;
0000   129 ;       3) New algorithm - special case y2=0
0000   130 ;
0000   131 ;       4) New algorithm - optimize branches according to probable
0000   132 ;       sign of R(H).
0000   133 ;
0000   134 ;
0000   135 ;
0000   136 ;
0000   137 ; variable use:
0000   138 ;
0000   139 ;                 size
0000   140 ;                 in
0000   141 ;     variable    digits          use
0000   142 ;     --------    ------    -------------------------
0000   143 ;        y1         15      High order digits of divisor.
0000   144 ;        y2         16      Low order digits of divisor.
0000   145 ;        x          31      Initially dividend, thereafter
0000   146 ;                           remainders of successive divide
0000   147 ;                           operations.
0000   148 ;        z        py(ap)    Quotient.
0000   149 ;        z2         31      Temporarily holds trial low
0000   150 ;                           order digits of quotient.
0000   151 ;        t1         31      High order digits of the remainder.
0000   152 ;        t2         31      Holds the 15 low order digits of the
0000   153 ;                           46 digit remainder. 31 digits for
0000   154 ;                           possible later changes.
0000   155 ;        t3         16      Holds the low order digits of the
0000   156 ;                           remainder.
0000   157 ;        t4         31      Temporary used because packed instructions can
0000   158 ;                           not overlap their operands.
0000   159 ;
0000   160 ;
0000   161 ;
0000   162 ;
0000   163 ;
0000   164 ;
0000   165 ; register usage:
0000   166 ;
0000   167 ;     register                  use
0000   168 ;     --------    ----------------------------------------------
0000   169 ;        r6       a = additional digits of precision required
0000   170 ;        r7       stky(sp) which holds a copy of divisor
0000   171 ;        r8       py(ap) = precision of y
```

```
                        ;       r9      r = number of additional digits of the quotient
                        ;               that are to be found for next step
                        ;       r10     z(ap)
                        ;       r11     pz(ap) = precision of quotient
                        ;
                        ;
                        ;--
                        ;
                        ; stack offsets for work area
                        ;
                                $offset 0,,<-                   ;
                                <,16>,-                         ;x, 31 digits
                                <stky1,8>,-                     ;y1 15 digits
                                <stky2,9>,-                     ;y2 16 digits
                                <stkz2,16>,-                    ;z2 31 digits
                                <stkt1,16>,-                    ;t1 31 digits
                                <stkt2,16>,-                    ;t2 31 digits (15 digits used)
                                <stky,16>,-                     ;y  31 digits
                                <stkt3,9>,-                     ;t3 16 digits
                                <stkt4,16>,-                    ;t4 31 digits
                                <stksign,1>,-                   ;sign of quotient, 2 bits
                                <stklen,0>,-                    ;length of work area
                                >                               ;
                        stky1:
                        stky2:
                        stkz2:
                        stkt1:
                        stkt2:
                        stky:
                        stkt3:
                        stkt4:
                        stksign:
                        stklen:
                        ;
                        ; parameter offsets
                        ;
                                $offset 4,,<-                   ;
                                <x>,-                           ;x = dividend by reference
                                <y>,-                           ;y = divisor by reference
                                <py>,-                          ;prec(y) by value
                                <z>,-                           ;z = quotient by reference
                                <pz>,-                          ;prec(z) by value
                                <consta>,-                      ;a as defined above
                                >                               ;
                        x:
                        y:
                        py:
                        z:
                        pz:
                        consta:
                        ;
                        ;       psect declarations
                        ;
00000000                        .psect _ots$code pic, usr, con, rel, lcl, shr, -
                                                exe, rd, nowrt, long
                        ;
                        ; constant data area
```

```
                                        0000   213 ;
                                     1C 0000   214 one:    .packed +1                          ;
                                     0C 0001   215 zero:   .packed +0                          ;
                                        0002   216 ;**warning** the following two data definitions must be contiguous
                                     09 0002   217 nines:  .byte   9                           ;10**16 - 1 (must be followed by 10**15-1)
                                  9C 99 0003   218 nine15: .packed +999999999999999          ;10**15 - 1
99 99 99 99 09 00 00 00 00 00 00 00 000B   219 bignine:.packed +00000000000000099999999999999999 ;10**16 - 1
                                  9C 99 0017
                                  00 00 001B   220 ten15:  .packed +1000000000000000         ;10**15
                                  9D 99 0024   221 neg9:   .packed -9999999999999999         ;-(10**16 - 1)
                                     002D   222 ;
                                     002D   223 ; local symbol definitions
                                     002D   224 ;
                               0000000F   002D   225 bytes_to_sign=15                          ;bytes to sign for fixed decimal 31
                                     002D   226 ;
                                     002D   227 ; run time routine ots$div_pk_long
                                     002D   228 ;
                                  CFFC 002D   229         .entry  ots$div_pk_long,^M<iv,dv,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11>
                           5E   85 AE   9E 002F   230         movab   -stklen(sp),sp              ;make room for temporaries
                           56   18 AC   D0 0033   231         movl    consta(ap),r6               ;get value of a
                           57   51 AE   9E 0037   232         movab   stky(sp),r7                 ;address of copy of divisor
                           58   0C AC   D0 003B   233         movl    py(ap),r8                   ;precision of divisor
                           5A   10 AC   D0 003F   234         movl    z(ap),r10                   ;save address of quotient
                           5B   14 AC   D0 0043   235         movl    pz(ap),r11                  ;precision of quotient
                                7A AE   94 0047   236         clrb    stksign(sp)                 ;clear sign flag
                        6E   04 BC   1F 34 004A   237         movp    #31,ax(ap),(sp)             ;move x, set cond. code
                                     24 14 004F   238         bgtr    50$                         ;branch if x > 0
                                     1C 19 0051   239         blss    40$                         ;branch if x < 0
                                        0053   240 ;
                                        0053   241 ;x = 0
                                        0053   242 ;
                     08 BC   58   AA AF   00 37 0053   243         cmpp4   #0,zero,r8,@y(ap)           ;divisor zero?
                                     09 13 005A   244         beql    30$                         ;branch if divide by 0
6A   5B   00   A0 AF   00   00   F8 005C   245         ashp    #0,#0,zero,#0,r11,(r10) ;z = 0
                                     04 0064   246         ret
  5B   95 AF   00   98 AF   00 27 0065   247 30$:    divp    #0,zero,#0,zero,r11,(r10) ;cause divide by zero
                                  6A 006D
                                     04 006E   248         ret
                                        006F   249 ;
                                        006F   250 ;x not 0
                                        006F   251 ;
                                        006F   252 40$:                                         ;
                                  7A AE   96 006F   253         incb    stksign(sp)                 ;set low order bit
                                  0F AE   97 0072   254         decb    bytes_to_sign(sp)           ;x < 0 so make it positive
                                        0075   255 ;
                                        0075   256 ;determine sign of y
                                        0075   257 ;y may be 0 at this point
                                        0075   258 ;optimized for y>0
                                        0075   259 ;
                                        0075   260 50$:                                         ;
                        67   08 BC   58 34 0075   261         movp    r8,@y(ap),(r7)              ;move y into temporary
                                     0D 18 007A   262         bgeq    60$                         ;branch if so
                                  7A AE   96 007C   263         incb    stksign(sp)                 ;set neg indicator
58   FF7A CF   00   08 BC   58 23 007F   264         subp6   r8,@y(ap),#0,zero,r8,(r7) ;convert to positive
                                     67 0088
                                        0089   265 ;
                                        0089   266 ;get y1 and y2
```

```
                                        0089   267 ;
                                        0089   268 60$:                                                  ;
0F 00 67 58 F0 8F F8                    0089   269         ashp    #-16,r8,(r7),#0,#15,stky1(sp) ;high order 15 digits of y
                  10 AE                 0090
1F 00 10 AE 0F 10 F8                    0092   270         ashp    #16,#15,stky1(sp),#0,#31,stkt1(sp) ;y with 16 low order zeroed
                  31 AE                 0099
10 67 58 31 AE 1F 23                    009B   271         subp6   #31,stkt1(sp),r8,(r7),#16,stky2(sp) ;16 low order digits y
                  18 AE                 00A2
                                        00A4   272 ;
                                        00A4   273 ;prec(y) is large enough for y1 to possibly not be 0
                                        00A4   274 ;
10 AE 0F FF58 CF 00 37                  00A4   275         cmpp4   #0,zero,#15,stky1(sp)   ;y1=0?
                  03 13                 00AC   276         beql    80$                     ;branch if y1 is zero
                0098 31                 00AE   277         brw     200$                    ;branch if y1 is not zero
                                        00B1   278 ;
                                        00B1   279 ; y2(16) holds all of y
                                        00B1   280 ;
                                        00B1   281 80$:                                                 ;
18 AE 10 6E 1F 37                       00B1   282         cmpp4   #31,(sp),#16,stky2(sp)  ;x<y ?
                  13 19                 00B7   283         blss    95$                     ;branch if x<y; shift x by 15 is ok
6A 5B 6E 1F 18 AE 10 27                 00B9   284         divp    #16,stky2(sp),#31,(sp),r11,(r10) ;z=x/y2
1F 6A 5B 18 AE 10 25                    00C1   285         mulp    #16,stky2(sp),r11,(r10),#31,stkt1(sp) ;t1=(x/y2)*y2
                  31 AE                 00C8
                  15 11                 00CA   286         brb     110$                    ;
5B 00 FF2F CF 00 00 F8                  00CC   287 95$:     ashp    #0,#0,zero,#0,r11,(r10) ;clear quotient
                  6A                    00D4
                  12 11                 00D5   288         brb     115$
1F 21 AE 0F 18 AE 10 25                 00D7   289 100$:    mulp    #16,stky2(sp),#15,stkz2(sp),#31,stkt1(sp) ;t1=y2*z2
                  31 AE                 00DF
6E 1F 31 AE 1F 22                       00E1   290 110$:    subp4   #31,stkt1(sp),#31,(sp)  ;x=x-t1
                  3E 13                 00E7   291         beql    150$                    ;branch if remainder = 0
                                        00E9   292 ;
                                        00E9   293 ;determine r, the number of the next low order digits to obtain
                                        00E9   294 ;
                  59 0F D0              00E9   295 115$:    movl    #15,r9                  ;r=15
                  59 56 D1              00EC   296         cmpl    r6,r9                   ;a>15?
                     03 18              00EF   297         bgeq    130$                    ;branch if larger
                  59 56 D0              00F1   298         movl    r6,r9                   ;r=a
31 AE 1F 00 6E 1F 59 F8                 00F4   299 130$:    ashp    r9,#31,(sp),#0,#31,stkt1(sp) ;shift x left by r
                     6E 31 AE 1F 34     00FC   300         movp    #31,stkt1(sp),(sp)      ;copy back into x
31 AE 5B 00 6A 5B 59 F8                 0101   301         ashp    r9,r11,(r10),#0,r11,stkt1(sp) ;shift z left by r
                        6A 31 AE 5B 34  0109   302         movp    r11,stkt1(sp),(r10)     ;copy back into z
   0F 6E 1F 18 AE 10 27                 010E   303         divp    #16,stky2(sp),#31,(sp),#15,stkz2(sp) ;z2(15)=x/y2
                     21 AE              0115
6A 5B 21 AE 0F 20                       0117   304         addp4   #15,stkz2(sp),r11,(r10) ;z=z+z2
                  56 59 C2              011D   305         subl2   r9,r6                   ;a=a-r
                     B5 12              0120   306         bneq    100$                    ;branch if more
                  13 7A AE E8           0122   307 140$:    blbs    stksign(sp),155$        ;branch if quotient <0
                     04                 0126   308         ret                             ;
                                        0127   309 ;
                                        0127   310 ;remainder = 0
                                        0127   311 ;
                                        0127   312 150$:                                                 ;remainder = 0
31 AE 5B 00 6A 5B 56 F8                 0127   313         ashp    r6,r11,(r10),#0,r11,stkt1(sp) ;account for scale
                  0B 7A AE E8           012F   314         blbs    stksign(sp),160$        ;branch if quotient is negative
                        6A 31 AE 5B 34  0133   315         movp    r11,stkt1(sp),(r10)     ;copy back into quotient
                     04                 0138   316         ret                             ;
```

```
                31 AE    6A    5B    34  0139   317 155$:   movp    r11,(r10),stkt1(sp)     ;copy quotient into temporary
                                       013E   318 160$:                                   ;enter if t1 holds quotient
5B    FEBB CF    00    31 AE    5B    23  013E   319         subp6   r11,stkt1(sp),#0,zero,r11,(r10) ;make z negative
                                 6A          0147
                                       04  0148   320         ret
                                           0149   321 ;
                                           0149   322 ;New division algorithm
                                           0149   323 ;
                                           0149   324 200$:                                 ;
                                           0149   325 ;
                                           0149   326 ;insure that we have x<y
                                           0149   327 ;
            67    58    6E    1F    37  0149   328         cmpp4   #31,(sp),r8,(r7)         ;x<y?
                                       19  19  014E   329         blss    220$                ;branch if x < y
   6A    5B    6E    1F    67    58    27  0150   330         divp    r8,(r7),#31,(sp),r11,(r10) ;z=x/y
31 AE    1F    67    58    6A    5B    25  0157   331         mulp    r11,(r10),r8,(r7),#31,stkt1(sp);t1=y*z
         6E    1F    31 AE    1F    22  015F   332         subp4   #31,stkt1(sp),#31,(sp)  ;x=x-t1
                                       C0  13  0165   333         beql    150$                ;branch if remainder is zero
                                       09  11  0167   334         brb     230$                ;
   5B    00    FE92 CF    00    00    F8  0169   335 220$:   ashp    #0,#0,zero,#0,r11,(r10) ;clear quotient
                                 6A          0171
                                           0172   336 ;
                                           0172   337 ;Assumes z is defined
                                           0172   338 ;       0 < x < y
                                           0172   339 ;       and r6=a gives the number of additional digits required.
                                           0172   340 ;Determine r = # of digits for next part of quotient.
                                           0172   341 ;
                     59    56    D0  0172   342 230$:   movl    r6,r9                   ;r=a
                     0F    56    D1  0175   343         cmpl    r6,#15                  ;a>15?
                                       03  15  0178   344         bleq    240$                ;branch if larger
                     59    0F    D0  017A   345         movl    #15,r9                  ;r=15
                     56    59    C2  017D   346 240$:   subl2   r9,r6                   ;update r6 = a
                                           0180   347 ;
                                           0180   348 ;calculate z2 = min(B-1,[x/y1])
                                           0180   349 ;
                                           0180   350 250$:                                 ;
   1F    6E    1F    10 AE    0F    27  0180   351         divp    #15,stky1(sp),#31,(sp),#31,stkt1(sp);t1=x/y1
                                    31 AE      0187
   FE71 CF    10    31 AE    1F    37  0189   352         cmpp4   #31,stkt1(sp),#16,nines ;t1>10**16-1?
                                       09  15  0191   353         bleq    310$                ;branch if not
         21 AE    FE6B CF    0F    34  0193   354         movp    #15,nine15,stkz2(sp)    ;move in base-1
                                       0A  11  019A   355         brb     320$                ;skip ashp, you have 15 digits
                                           019C   356 ;
                                           019C   357 ;calculate y2*z2
                                           019C   358 ;       t3 = high order 16 digits of y2*z2
                                           019C   359 ;       t2 =  low order 15 digits of y2*z2
                                           019C   360 ;
                                           019C   361 310$:
0F    00    31 AE    1F    FF 8F    F8  019C   362         ashp    #-1,#31,stkt1(sp),#0,#15,stkz2(sp) ;we only get 15 digits
                                    21 AE      01A4
1F    18 AE    10    21 AE    0F    25  01A6   363 320$:   mulp    #15,stkz2(sp),#16,stky2(sp),#31,stkt1(sp) ;t1=y2*z2
                                    31 AE      01AE
10    00    31 AE    1F    F1 8F    F8  01B0   364         ashp    #-15,#31,stkt1(sp),#0,#16,stkt3(sp) ;t3(16)=t1 shifted right 15
                                    61 AE      01B8
   1F    00    61 AE    10    0F    F8  01BA   365         ashp    #15,#16,stkt3(sp),#0,#31,stkt4(sp) ;t4(31) = t3 shifted left 15
                                    6A AE      01C1
0F    31 AE    1F    6A AE    1F    23  01C3   366         subp6   #31,stkt4(sp),#31,stkt1(sp),#15,stkt2(sp) ;t2(15)=t1-t4
```

```
                     41 AE          01CB
                        4C    13    01CD    367          beql    330$                    ;branch if no borrow required
                                    01CF    368 ;
                                    01CF    369 ;borrow is -1, t2 not 0
                                    01CF    370 ;calculate R(H) =
                                    01CF    371 ;       t1(31) = 31 high order digits of x(46) - y*z2
                                    01CF    372 ;       t2(15) = 15  low order digits of x(46) - y*z2
                                    01CF    373 ;
  1E    10 AE    0F    21 AE    0F    25    01CF    374          mulp    #15,stkz2(sp),#15,stky1(sp),#30,stkt4(sp) ;t4(30)=y1*z2
                                 6A AE          01D7
     1F    00    6A AE    1E    01    F8    01D9    375          ashp    #1,#30,stkt4(sp),#0,#31,stkt1(sp);31 high order of 46
                              31 AE          01E0
       31 AE    1F    61 AE    10    20    01E2    376          addp4   #16,stkt3(sp),#31,stkt1(sp) ;t1(31)=t1(31)+t3(16)
     1F    6E    1F    31 AE    1F    23    01E9    377          subp6   #31,stkt1(sp),#31,(sp),#31,stkt4(sp) ;t4=x-t1
                                 6A AE          01F0
                                    1E    15    01F2    378          bleq    325$                    ;branch if R(H) negative (t4 < = 0)
1F    6A AE    1F    FE07 CF    01    23    01F4    379          subp6   #1,one,#31,stkt4(sp),#31,stkt1(sp) ;t1=t4-1 (borrow 1)
                              31 AE          01FD
0F    FE14 CF    10    41 AE    0F    23    01FF    380          subp6   #15,stkt2(sp),#16,ten15,#15,stkt4(sp) ;t4=10**15-t2
                                 6A AE          0208
              41 AE    6A AE    0F    34    020A    381          movp    #15,stkt4(sp),stkt2(sp) ;copy back into t2
                                 63    11    0210    382          brb     370$                    ;
              31 AE    6A AE    1F    34    0212    383 325$:    movp    #31,stkt4(sp),stkt1(sp) ;make t1 hold high order R(H)
                           00AF    31    0218    384          brw     500$                    ;R(H) < 0 (can not be zero)
                                    021B    385 ;
                                    021B    386 ;no borrow, t2 = 0
                                    021B    387 ;calculate R(H) = t1(31) = 31 high order digits of x(46) - y*z2
                                    021B    388 ;                 t2(15) = 0 = 15 low order digits of x(46) - y*z2
                                    021B    389 ;
                                    021B    390 330$:
  1E    10 AE    0F    21 AE    0F    25    021B    391          mulp    #15,stkz2(sp),#15,stky1(sp),#30,stkt1(sp) ;t1(30)=z2(15)*y1(15)
                                 31 AE          0223
     1F    00    31 AE    1E    01    F8    0225    392          ashp    #1,#30,stkt1(sp),#0,#31,stkt4(sp) ;31 high order 46
                                 6A AE          022C
       6A AE    1F    61 AE    10    20    022E    393          addp4   #16,stkt3(sp),#31,stkt4(sp) ;t4=t3(16)+t4
     1F    6E    1F    6A AE    1F    23    0235    394          subp6   #31,stkt4(sp),#31,(sp),#31,stkt1(sp) ;t1=x-t4
                              31 AE          023C
                                 35    12    023E    395          bneq    370$                    ;branch if remainder not zero
                                    0240    396 ;
                                    0240    397 ;*****************************************************************
                                    0240    398 ;*                                                               *
                                    0240    399 ;* R(H) = 0 , i.e.                                               *
                                    0240    400 ;*remainder is zero                                               *
                                    0240    401 ;*z2(15) holds last non-zero digits of the quotient               *
                                    0240    402 ;*quotient has not been shifted yet to recieve z2                 *
                                    0240    403 ;*                                                               *
                                    0240    404 ;* this is a local routine with no exit other than ret           *
                                    0240    405 ;*****************************************************************
                                    0240    406 ;
                                    0240    407 340$:
6A AE    5B    00    6A    5B    59    F8    0240    408          ashp    r9,r11,(r10),#0,r11,stkt4(sp) ;t4=z shifted left by r9=r
                 6A    6A AE    5B    34    0248    409          movp    r11,stkt4(sp),(r10)     ;copy back quotient
                                 59    0F    C2    024D    410          subl2   #15,r9                  ;shift needed to leave r9 digits
     0F    00    21 AE    0F    59    F8    0250    411          ashp    r9,#15,stkz2(sp),#0,#15,stkt1(sp);low order digits of z
                              31 AE          0257
           6A    5B    31 AE    0F    20    0259    412          addp4   #15,stkt1(sp),r11,(r10) ;z=z+z2
                                 56    D5    025F    413          tstl    r6                      ;a = 0 ?
```

```
                                52   13  0261    414          beql    395$                    ;branch if a=0 (last iteration)
31 AE   5B   00   6A   5B      56   F8  0263    415          ashp    r6,r11,(r10),#0,r11,stkt1(sp);adjust for scale
                          50 7A AE   E8  026B    416 350$:    blbs    stksign(sp),410$        ;branch if quotient < 0
                6A   31 AE   5B     34  026F    417          movp    r11,stkt1(sp),(r10)     ;back into quotient
                                04   0274    418          ret
                                53   19  0275    419 370$:    blss    500$                    ;branch if R(H)<0
                                     0277    420 ;
                                     0277    421 ;*************************************************************************
                                     0277    422 ;*R(H) > 0 .  i.e.                                                     *
                                     0277    423 ;* t1 > 0                                                           *
                                     0277    424 ;*                                                                  *
                                     0277    425 ;*************************************************************************
                                     0277    426 ;
6A AE   5B   00   6A   5B      59   F8  0277    427 380$:    ashp    r9,r11,(r10),#0,r11,stkt4(sp) ;t4=z shifted left by r9=r
                6A   6A AE   5B     34  027F    428          movp    r11,stkt4(sp),(r10)     ;copy back quotient
                                56   D5  0284    429          tstl    r6                      ;a = 0 ?
                                1B   13  0286    430          beql    390$                    ;branch if a=0 (last iteration)
                                     0288    431 ;
                                     0288    432 ;a not 0
                                     0288    433 ;
        6A   5B   21 AE   0F      20  0288    434          addp4   #15,stkz2(sp),r11,(r10) ;z=z+z2
     1F 00   31 AE   1F   0F      F8  028E    435          ashp    #15,#31,stkt1(sp),#0,#31,stkt4(sp) ;t4=t1 shifted left 15
                          6A AE              0295
  1F   41 AE   0F   6A AE   1F    21  0297    436          addp6   #31,stkt4(sp),#15,stkt2(sp),#31,(sp) ;x=t4+t2
                                6E   029F
                          FECF       31  02A0    437          brw     230$
                                     02A3    438 ;
                                     02A3    439 ;a = 0
                                     02A3    440 ;
                                59   0F  C2  02A3    441 390$:    subl2   #15,r9                  ;shift needed to leave r9 digits
     0F 00   21 AE   0F   59      F8  02A6    442          ashp    r9,#15,stkz2(sp),#0,#15,stkt1(sp);low order digits of z
                             31 AE           02AD
        6A   5B   31 AE   0F      20  02AF    443          addp4   #15,stkt1(sp),r11,(r10) ;z=z+z2
                          01 7A AE   E8  02B5    444 395$:    blbs    stksign(sp),400$        ;branch if quotient <0
                                04   02B9    445          rst                             ;
                                     02BA    446 400$:                                    ;
                31 AE   6A   5B      34  02BA    447          movp    r11,(r10),stkt1(sp)     ;make copy of quotient
5B   FD3A CF   00   31 AE   5B      23  02BF    448 410$:    subp6   r11,stkt1(sp),#0,zero,r11,(r10) ;make z negative
                             6A              02C8
                                04   02C9    449          ret
                                     02CA    450 ;
                                     02CA    451 ;*************************************************************************
                                     02CA    452 ;*                                                                  *
                                     02CA    453 ;*R(H) < 0                                                           *
                                     02CA    454 ;* so check if R(H) + Y > = 0                                       *
                                     02CA    455 ;*                                                                  *
                                     02CA    456 ;*************************************************************************
                                     02CA    457 ;
                                     02CA    458 500$:
     FD52 CF   10   31 AE   1F      37  02CA    459          cmpp4   #31,stkt1(sp),#16,neg9  ;t1< -(10**16-1) ?
                                66   19  02D2    460          blss    600$                    ;branch if R(H) + Y < 0
     1F   00   31 AE   1F   0F      F8  02D4    461          ashp    #15,#31,stkt1(sp),#0,#31,stkt4(sp) ;shift t1 left 15
                          6A AE              02DB
  1F   6A AE   1F   41 AE   0F      23  02DD    462          subp6   #15,stkt2(sp),#31,stkt4(sp),#31,stkt1(sp) ;t1=t1-t2
                             31 AE           02E5
        31 AE   1F   67      5B      20  02E7    463          addp4   r8,(r7),#31,stkt1(sp)   ;t1=t1+y
                                4B   19  02ED    464          blss    600$                    ;branch if R(H) + Y < 0
```

```
                                3E   13  02EF    465          beql    530$                            ;branch if no remainder (first adjust z2)
     21 AE   0F    FDOA CF     01   22  02F1    466          subp4   #1,one,#15,stkz2(sp)            ;z2=z2-1
6A AE   5B   00    6A   5B     59   F8  02F9    467          ashp    r9,r11,(r10),#0,r11,stkt4(sp) ;t4=z shifted left by r9=r
             6A    6A AE   5B   34   0301    468          movp    r11,stkt4(sp),(r10)             ;copy back quotient
                           56   D5  0306    469          tstl    r6                              ;a = 0 ?
                           0E   13  0308    470          beql    510$                            ;branch if a=0 (last iteration)
                                    030A    471 ;
                                    030A    472 ;a not 0
                                    030A    473 ;
          6A   5B   21 AE   0F   20  030A    474          addp4   #15,stkz2(sp),r11,(r10) ;z=z+z2
               6E   31 AE    1F   34  0310    475          movp    #31,stkt1(sp),(sp)      ;x=t1
                         FE5A   31  0315    476          brw     230$
                                    0318    477 ;
                                    0318    478 ;a = 0
                                    0318    479 ;
                     59   0F   C2  0318    480 510$:    subl2   #15,r9                          ;shift needed to leave r9 digits
     0F   00   21 AE   0F   59   F8  031B    481          ashp    r9,#15,stkz2(sp),#0,#15,stkt1(sp);low order digits of z
                           31 AE            0322
          6A   5B   31 AE   0F   20  0324    482          addp4   #15,stkt1(sp),r11,(r10) ;z=z+z2
                       8C 7A AE   E8  032A    483          blbs    stksign(sp),400$        ;branch if quotient <0
                                04  032E    484          ret
                                    032F    485 ;
                                    032F    486 ;remainder is zero
                                    032F    487 ;
     21 AE   0F    FCCC CF     01   22  032F    488 530$:    subp4   #1,one,#15,stkz2(sp)            ;z2=z2-1
                         FF06   31  0337    489          brw     340$                            ;go add to quotient, then ret
                                    033A    490 ;
                                    033A    491 ;*****************************************************************************
                                    033A    492 ;*                                                                           *
                                    033A    493 ;*R(H) + Y < 0                                                              *
                                    033A    494 ;*calculate L = min(B-1,[X/(y1+1)])                                         *
                                    033A    495 ;* by theorem 5, L=[X/(y1+1)]                                              *
                                    033A    496 ;*                                                                           *
                                    033A    497 ;*****************************************************************************
                                    033A    498 ;
10   10 AE   0F    FCC1 CF     01   21  033A    499 600$:    addp6   #1,one,#15,stky1(sp),#16,stkt3(sp) ;t3=y1+1
                           61 AE            0343
     10   6E   1F    61 AE    10   27  0345    500          divp    #16,stkt3(sp),#31,(sp),#16,stkt1(sp) ;t1=x/t3
                           31 AE            034C
                                    034E    501 ;
                                    034E    502 ;split up y2*L
                                    034E    503 ;        t3 = high order 16 digits of y2*L
                                    034E    504 ;        t2 = low  order 15 digits of y2*L
  0F   00   31 AE   10    FF 8F    F8  034E    505          ashp    #-1,#16,stkt1(sp),#0,#15,stkz2(sp) ;we only get 15 digits
                           21 AE            0356
  1F   18 AE   10   21 AE    0F   25  0358    506          mulp    #15,stkz2(sp),#16,stky2(sp),#31,stkt1(sp) ;t1=y2*z2
                           31 AE            0360
  10   00   31 AE   1F    F1 8F    F8  0362    507          ashp    #-15,#31,stkt1(sp),#0,#16,stkt3(sp) ;t3(16)=t1 shifted right 15
                           61 AE            036A
     1F   00   61 AE   10    0F    F8  036C    508          ashp    #15,#16,stkt3(sp),#0,#31,stkt4(sp) ;t4(31) = t3 shifted left 15
                           6A AE            0373
  0F   31 AE   1F   6A AE    1F   23  0375    509          subp6   #31,stkt4(sp),#31,stkt1(sp),#15,stkt2(sp) ;t2(15)=t1-t4
                           41 AE            037D
                                    037F    510 ;
                                    037F    511 ;*****************************************************************************
                                    037F    512 ;*                                                                           *
                                    037F    513 ;* calculate R(L)                                                           *
```

```
                                        037F    514 ;*                                                                      *
                                        037F    515 ;*******************************************************************************
                                        037F    516 ;
                                    41  13  037F    517         beql    630$                    ;branch if no borrow required
                                        0381    518 ;
                                        0381    519 ;borrow is -1, t2 not 0
                                        0381    520 ;calculate R(L) =
                                        0381    521 ;       t1(31) = 31 high order digits of x(46) - y*z2
                                        0381    522 ;       t2(15) = 15  low order digits of x(46) - y*z2
                                        0381    523 ;note: it is always true that R(L) > = 0
                                        0381    524 ;
0F  FC92 CF  10  41 AE  0F      23      0381    525         subp6   #15,stkt2(sp),#16,ten15,#15,stkt4(sp) ;t4=10**15-t2
                                6A AE           038A
          41 AE  6A AE  0F      34      038C    526         movp    #15,stkt4(sp),stkt2(sp) ;copy back into t2
  1E  10 AE  0F  21 AE  0F      25      0392    527         mulp    #15,stkz2(sp),#15,stky1(sp),#30,stkt4(sp) ;t4(30)=y1*z2
                                6A AE           039A
    1F  00  6A AE  1E  01       F8      039C    528         ashp    #1,#30,stkt4(sp),#0,#31,stkt1(sp) ;high order 31 of 46
                                31 AE           03A3
      31 AE  1F  61 AE  10      20      03A5    529         addp4   #16,stkt3(sp),#31,stkt1(sp) ;t1(31)=t1(31)+t3(16)
    1F  6E  1F  31 AE  1F       23      03AC    530         subp6   #31,stkt1(sp),#31,(sp),#31,stkt4(sp) ;t4=x-t1
                                6A AE           03B3
1F  6A AE  1F  FC46 CF  01      23      03B5    531         subp6   #1,one,#31,stkt4(sp),#31,stkt1(sp) ;t1=t4-1 (borrow 1)
                                31 AE           03BE
                                    23  11  03C0    532         brb     700$                    ;
                                        03C2    533 ;
                                        03C2    534 ;no borrow, t2 = 0
                                        03C2    535 ;calculate R(L) = t1(31) = x - y1*z2
                                        03C2    536 ;
                                        03C2    537 630$:
  1E  10 AE  0F  21 AE  0F      25      03C2    538         mulp    #15,stkz2(sp),#15,stky1(sp),#30,stkt1(sp) ;t1(30)=y1*z2
                                31 AE           03CA
    1F  00  31 AE  1E  01       F8      03CC    539         ashp    #1,#30,stkt1(sp),#0,#31,stkt4(sp) ;high order 31 of 46
                                6A AE           03D3
      6A AE  1F  61 AE  10      20      03D5    540         addp4   #16,stkt3(sp),#31,stkt4(sp) ;t4(31)=t4(31)+t3(16)
    1F  6E  1F  6A AE  1F       23      03DC    541         subp6   #31,stkt4(sp),#31,(sp),#31,stkt1(sp) ;t1=x-t4
                                31 AE           03E3
                                        03E5    542 ;
                                        03E5    543 ;*******************************************************************************
                                        03E5    544 ;*                                                                             *
                                        03E5    545 ;* calculate Z2 = L + R(L)/Y                                                   *
                                        03E5    546 ;*                                                                             *
                                        03E5    547 ;*******************************************************************************
                                        03E5    548 ;
    1F  00  31 AE  1F  0F       F8      03E5    549 700$:   ashp    #15,#31,stkt1(sp),#0,#31,stkt4(sp) ;shift t1 left 15
                                6A AE           03EC
  6A AE  1F  41 AE  0F      20      03EE    550         addp4   #15,stkt2(sp),#31,stkt4(sp) ;t4=t4+t2
0F  6A AE  1F  67  58       27      03F5    551         divp    r8,(r7),#31,stkt4(sp),#15,stkt1(sp) ;t1=t4/y
                                31 AE           03FC
  21 AE  0F  31 AE  0F      20      03FE    552         addp4   #15,stkt1(sp),#15,stkz2(sp) ;z2=z2+t1
                            FD9E    31  0405    553         brw     320$                    ;
                                        0408    554
                                        0408    555         .end
```

```
BIGNINE             0000000B R      02
BYTES_TO_SIGN   =   0000000F
CONSTA              00000018
DIR...          =   00000001
NEG9                00000024 R      02
NINE15              00000003 R      02
NINES               00000002 R      02
ONE                 00000000 R      02
OTS$DIV_PK_LONG     0000002D RG     02
PY                  0000000C
PZ                  00000014
STKLEN              0000007B
STKSIGN             0000007A
STKT1               00000031
STKT2               00000041
STKT3               00000061
STKT4               0000006A
STKY                00000051
STKY1               00000010
STKY2               00000018
STKZ2               00000021
TEN15               0000001B R      02
X                   00000004
Y                   00000008
Z                   00000010
ZERO                00000001 R      02
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	0000007B (123.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_OTS$CODE	00000408 (1032.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:00.61
Command processing	113	00:00:00.33	00:00:01.93
Pass 1	139	00:00:01.52	00:00:06.50
Symbol table sort	0	00:00:00.03	00:00:00.04
Pass 2	119	00:00:00.82	00:00:03.32
Symbol table output	4	00:00:00.01	00:00:00.02
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	409	00:00:02.79	00:00:12.44

```
The working set limit was 1050 pages.
15986 bytes (32 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 27 non-local and 37 local symbols.
555 source lines were read in Pass 1, producing 14 object records in Pass 2.
```

3 pages of virtual memory were used to define 2 macros.

+----------------------------+
! Macro library statistics !
+----------------------------+

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	2

45 GETS were required to define 2 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:OTSPKDIVL/OBJ=OBJ$:OTSPKDIVL MSRC$:OTSPKDIVL/UPDATE=(ENH$:OTSPKDIVL)

OTSPKDIVL
LIS

OTSPKDIVS
LIS

STRABMUL
LIS

STRCHESTA
LIS

RTLLIB
LIS

STRAPPEND
LIS

STRARITH
LIS

STRALLOC
LIS

STRANASTR
LIS

STRCOMCAS
LIS

OTSSCOPY
LIS

OTSTERMIO
LIS