

```
; Mike Gonta, 2019
; look and see - many look but few see

; PDFasm - assembles a copy of it's own source to a PDF
; Select all the text in a PDF viewer and paste into a text file
; Usage: fasmg PDFasm.asm
```

```
LINE_LENGTH equ 80
NUMBER_OF_LINES equ 25
```

```
if defined PDF
    this = PDF
else
    this = __FILE__
end if
```

```
virtual at 0
input:: file this
    .length = $
end virtual
```

```
repeat input.length - 1, i:0
    load a:byte from input:i
    if a = 0x0A
        EOL equ 0x0A
```

```

        break
    else if a = 0x0D
        load a:byte from input:i + 1
        if a = 0x0A
            EOL equ 0x0D, 0x0A
        else
            EOL equ 0x0D
        end if
        break
    end if
end repeat

virtual at 0
    db EOL
    if $ = 1
        EOL_2_BYTES equ ' ', EOL
    else
        EOL_2_BYTES equ EOL
    end if
end virtual

macro to_number number, leading_zeros = 0
    num = number
    divisor = 1000000000
    leader = leading_zeros

```

```
while divisor > 1
  result = num / divisor
  leader = leader OR result
  num = num MOD divisor
  divisor = divisor / 10
  if leader <> 0
    db '0' + result
  end if
end while
db '0' + num
end macro
```

postpone ?

```
number_of_pages = 0
objects = 0
warning = 0
```

```
if this = __FILE__
  format binary as 'pdf'
  db '%PDF-1.0', EOL
end if
```

```
count = input.length
position = 0
```

```
objects = objects + 1
font_object = objects
font = $%
to_number font_object
db ' 0 obj', EOL
db '<<', EOL
db '/Type /Font', EOL
db '/Subtype /TrueType', EOL
db '/Name /F1', EOL
db '/BaseFont /Courier', EOL
db '/Encoding /WinAnsiEncoding', EOL
db '>>', EOL
db 'endobj', EOL
```

```
source_line = 1
page_number = objects + 1
repeat 65536 ; objects
  objects = objects + 1
  number_of_pages = number_of_pages + 1
  content#% = $%
  to_number objects
  db ' 0 obj', EOL
  db '<</Length '
  objects = objects + 1
  to_number objects ; indirect reference after this object
```

```
db ' 0 R>>', EOL
db 'stream', EOL
stream#%:
db 'BT', EOL
db '/F1 15 Tf', EOL
db '36 506 Td', EOL
db '16 TL', EOL
lines = 1
while lines <= NUMBER_OF_LINES ; pages
  old_position = position
  db '('
  if count > LINE_LENGTH
    count = LINE_LENGTH
  end if
  while count > 0 ; lines
    load a:byte from input:position
    if a = 0x0A | a = 0x0D
      break
    end if
    if a = '\\' | a = '(' | a = ')'
      db '\\'
    end if
    db a
    position = position + 1
    count = count - 1
  end while
  lines = lines + 1
end while
```

```
end while
if old_position = position ; blank line
  db ' ' ; add a space to the blank line so that text copied from the pdf
end if ; also has the blank line
db ")'", EOL ; end of the line
count = input.length - position
while count > 0 ; gets rid of EOL if any or truncates long lines
  load a:byte from input:position
  position = position + 1
  if a = 0x0A
    count = count - 1
    break
  end if
  if a = 0x0D
    count = count - 1
    if count = 0
      break
    end if
  load a:byte from input:position
  if a = 0x0A
    position = position + 1
    count = count - 1
    break
  end if
  break
end if
break
```

```
    end if
    count = count - 1
    warning = source_line
end while
if warning <> 0
    repeat 1, the_line:warning
        display 'WARNING: line #', `the_line, ' truncated.', EOL
    end repeat
end if
warning = 0
if count = 0
    break
end if
source_line = source_line + 1
lines = lines + 1
end while
db 'ET', EOL
endstream#%:
db 'endstream', EOL
db 'endobj', EOL
length#% = $%
to_number objects
db ' 0 obj', EOL
to_number endstream#% - stream#%
db EOL
```

```
db 'endobj', EOL
if count = 0
  break
end if
end repeat
```

```
root = $%
objects = objects + 1
root_object = objects
to_number root_object
db ' 0 obj', EOL
db '<<', EOL
db '/Type /Catalog', EOL
db '/Pages '
objects = objects + 1
pages_object = objects
to_number pages_object
db ' 0 R', EOL
db '>>', EOL
db 'endobj', EOL
pages = $%
to_number pages_object
db ' 0 obj', EOL
db '<<', EOL
db '/Type /Pages', EOL
```



```
db '/Count '  
to_number number_of_pages  
page_object = objects + 1  
db EOL  
db '/Kids ['  
repeat number_of_pages  
  to_number page_object  
  db ' 0 R '  
  page_object = page_object + 1  
end repeat  
db ']', EOL  
db '>>', EOL  
db 'endobj', EOL
```

```
repeat number_of_pages  
  objects = objects + 1  
  page#% = $%  
  to_number objects  
  db ' 0 obj', EOL  
  db '<<', EOL  
  db '/Type /Page', EOL  
  db '/Parent '  
  to_number pages_object  
  db ' 0 R', EOL  
  db '/MediaBox [0 0 792 612]', EOL ; landscape - 792/72 = 11, 612/72 = 8.5
```

```
db '/Resources <</Font <</F1 '  
to_number font_object  
db ' 0 R>> /ProcSet [/PDF /Text]>>', EOL  
db '/Contents '  
to_number page_number  
db ' 0 R', EOL  
db '>>', EOL  
db 'endobj', EOL  
page_number = page_number + 2  
end repeat
```

```
xref = $%  
db 'xref', EOL  
objects = objects + 1  
db '0 '  
to_number objects  
db EOL, '0000000000 65535 f', EOL_2_BYTES  
to_number font, 1  
db ' 0000 n', EOL_2_BYTES  
repeat number_of_pages  
  to_number content#%, 1  
  db ' 0000 n', EOL_2_BYTES  
  to_number length#%, 1  
  db ' 0000 n', EOL_2_BYTES  
end repeat
```

```
to_number root, 1
db ' 00000 n', EOL_2_BYTES
to_number pages, 1
db ' 00000 n', EOL_2_BYTES
repeat number_of_pages
  to_number page#%, 1
  db ' 00000 n', EOL_2_BYTES
end repeat
```

```
db 'trailer', EOL
db '<<', EOL
db '/Root '
to_number root_object
db ' 0 R', EOL
db '>>', EOL
db 'startxref', EOL
to_number xref
db EOL
db '%%EOF', EOL
```

```
end postpone
```