

geolink Internals

ShadowM

C=4 Expo

May 28, 2010

agenda

- ✓ build process
- ✓ GEOS for programmers
- ✓ geolink implementation
- ✓ example geoDebugger session

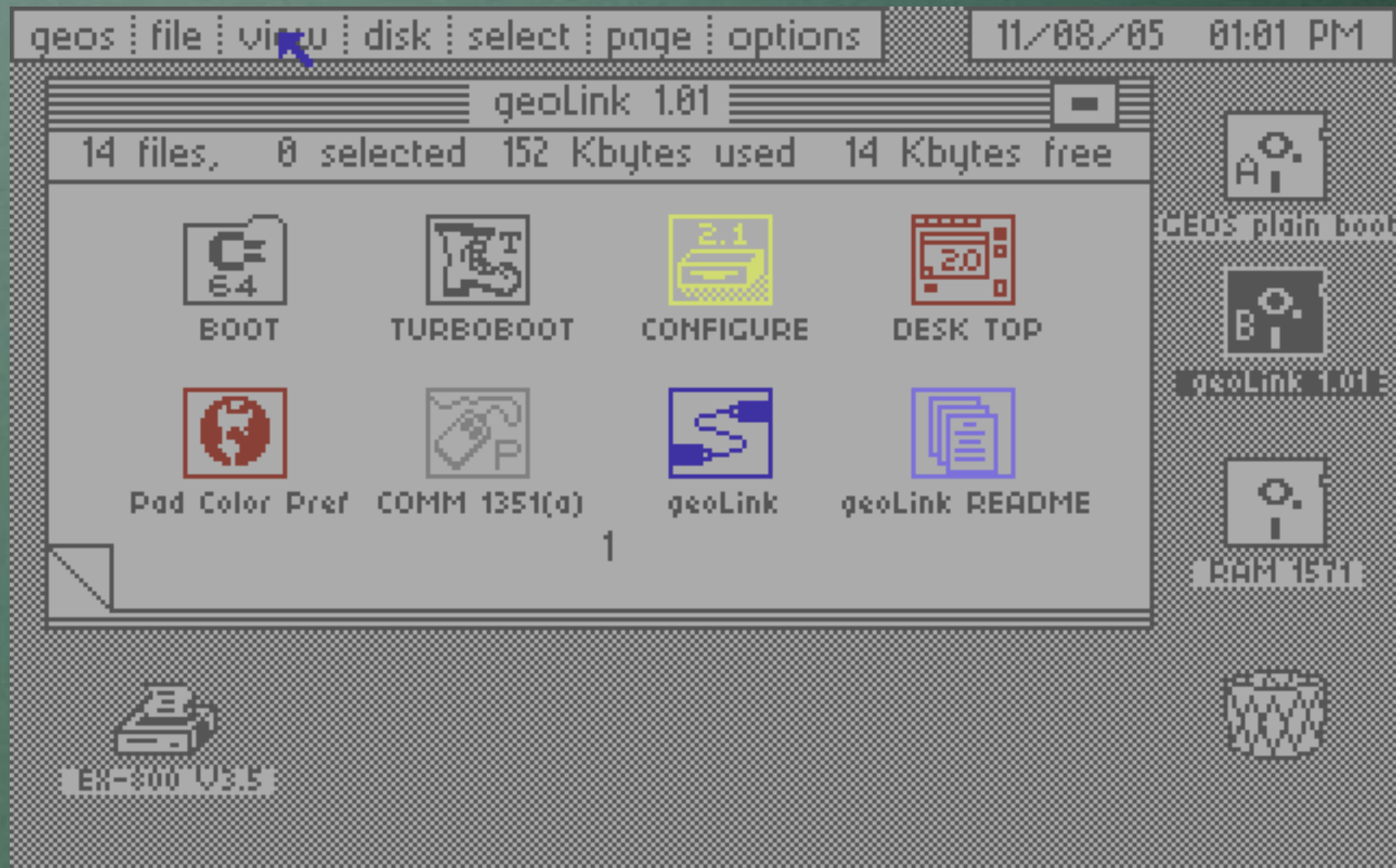
build process

- ✓ geolink: coded using geoWrite and assembled with geoProgrammer (no cross assemblers!)
- ✓ ipb5: slightly modified build from Jonno's source (built with cc65 toolchain)
- ✓ monospace font: created using geoFont
- ✓ standalone program to embed IP stack and font into geolink (also written with geoProgrammer)

GEOS for programmers

- ✓ GEOS file system
- ✓ GEOS programming conventions
- ✓ lifecycle of a GEOS program
- ✓ GEOS API overview
- ✓ memory map, I/O settings

GEOS deskTop



GEOS file system

- ✓ USR files; extra directory bytes are used
- ✓ two pointers in directory entry:
 - x pointer to file's data
 - x pointer to file header block (icon, description)
- ✓ VLIR files
 - x data pointer points to a single sector containing links to up to 127 sector chains

GEOS directory bytes

```
$8048 46 49 47 55 52 45 A0 A0 FIGURE
$8050 A0 A0 A0 A0 A0 05 14 01
$8058 0E 58 07 06 0D 03 59 00 .X...Y.
$8060 00 00 83 09 0F 44 45 53 .X...DES
$8068 4B 20 54 4F 50 A0 A0 A0 K TOP
$8070 A0 A0 A0 A0 A0 09 07 01
$8078 04 58 08 13 0D 23 78 00 .X...#x.
      +0 +1 +2 +3 +4 +5 +6 +7 01234567
$8080 00 00 83 0F 06 50 61 64 .Color Pad
$8088 20 43 6F 6C 6F 72 20 50 Color P
$8090 72 65 66 A0 A0 0F 13 00 ref ...
$8098 04 5B 0C 10 14 13 07 01 ...
$80A0 00 00 83 0F 07 43 4F 4D ...COM
$80B0 40 20 31 33 35 31 28 61 M 1351(a
$80B8 20 A0 A0 0F 14 00 00 .X...g
$80C0 00 00 83 0F 08 67 65 6F ...geo
$80C8 4C 69 6E 6B A0 A0 A0 Link
$80D0 A0 A0 A0 A0 0F 00 01
$80D8 06 0A 04 1A 09 0D 64 00 ...d.
$80E0 00 00 83 15 04 67 65 6F ...geo
$80E8 4C 69 6E 6B 20 52 45 41 Link REA
$80F0 44 4D 45 A0 A0 15 0F 01 DME ...
$80F8 07 0A 04 20 0F 00 .....-..
y
```

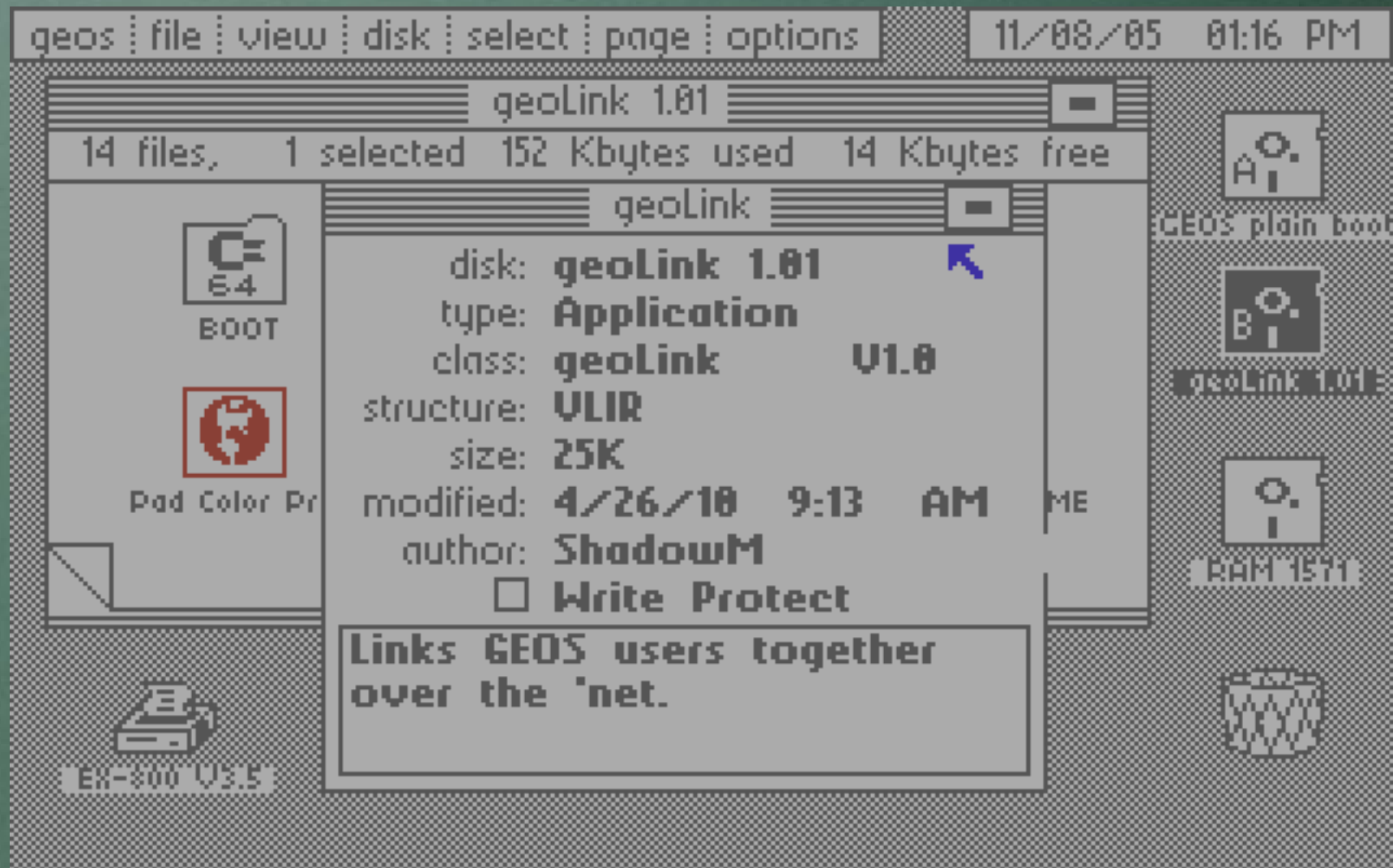
Annotations:

- GEOS file type**: points to the first byte of the file type field (06).
- header block pointer**: points to the first byte of the header block pointer field (0F).
- VLIR flag**: points to the first byte of the VLIR flag field (01).
- timestamp**: points to the first byte of the timestamp field (0A).

header block

```
$8010 FF 80 03 FF 80 1E FF 81 . . . . .
$8018 E3 FF 82 00 FF 82 00 01 c . . . . .
$8020 81 E0 01 80 1C 01 80 03 . . . . .
$8028 81 80 00 41 FF 00 41 FF . . . A . . A .
$8030 C3 81 FF 7C 01 FF C0 01 C . . l . . c .
$8038 FF 00 01 80 00 01 80 00 . . . . .
$8040 01 FF FF FF 83 06 01 00 . . . . .
$8048 04 FF 03 00 04 67 65 6F . . . . . geo
$8050 4C 69 6E 6B 20 20 20 20 Link
$8058 20 56 31 2E 30 00 00 00 U1.0 . . .
$8060 00 53 68 61 64 6F 77 4D .ShadowM
$8068 20 20 20 20 20 20 20 20
$8070 20 20 20 20 00 00 00 00 . . . .
$8078 00 00 00 00 00 00 00 00
      +0 +1 +2 +3 +4 +5 +6 +7 01234567
$8080 00 00 00 00 00 00 00 00 . . . . .
$8088 00 00 00 00 00 00 00 00 . . . . .
$8090 00 00 00 00 00 00 00 00 . . . . .
$8098 00 00 00 00 00 00 00 00 . . . . .
$80A0 4C 69 6E 6B 73 20 47 45 Links GE
$80A8 4F 53 20 75 73 65 72 73 OS users
$80B0 20 74 6F 67 65 74 68 65 togethe
$80B8 72 20 6F 76 65 72 20 74 r over t
$80C0 68 65 20 27 6E 65 74 2E he 'net.
--- <Space> or <Return> for more ---
```


info dialog (from header block)



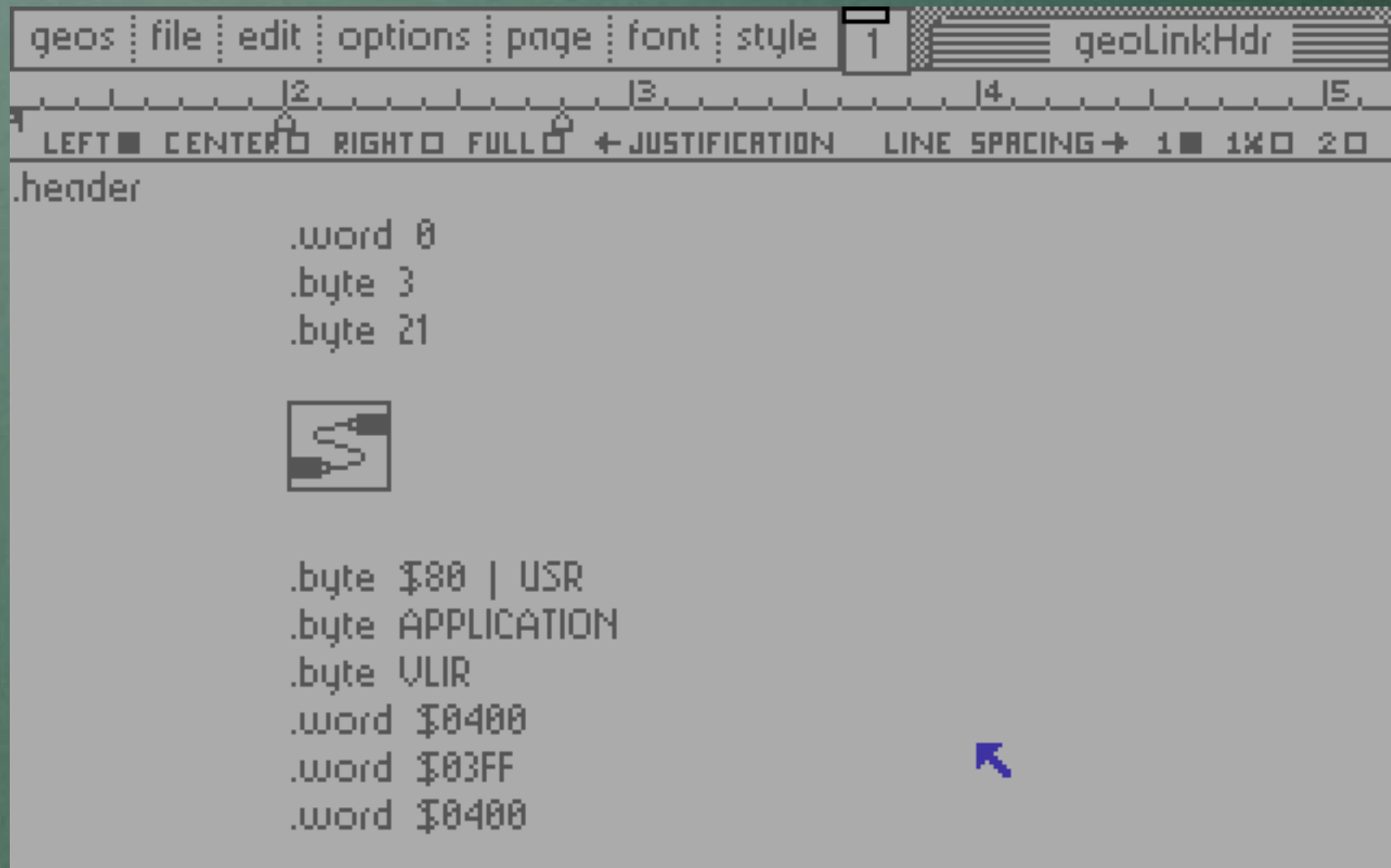
VLIW index block

```
>getb 0f,08
Block at: ($0F,$08)
+0 +1 +2 +3 +4 +5 +6 +7 01234567
$8000 00 FF 0F 10 0F 0D 10 0A .....
$8008 10 10 11 09 00 FF 00 FF .....
$8010 00 FF 13 01 13 06 00 FF .....
$8018 00 FF 00 FF 00 FF 00 FF .....
$8020 00 FF 00 FF 00 FF 00 FF .....
$8028 00 FF 00 FF 00 FF 00 FF .....
$8030 00 FF 00 FF 00 FF 00 FF .....
$8038 00 FF 00 FF 00 FF 00 FF .....
$8040 00 FF 00 FF 00 FF 00 FF .....
$8048 00 FF 00 FF 00 FF 00 FF .....
$8050 00 FF 00 FF 00 FF 00 FF .....
$8058 00 FF 00 FF 00 FF 00 FF .....
$8060 00 FF 00 FF 00 FF 00 FF .....
$8068 00 FF 00 FF 00 FF 00 FF .....
$8070 00 FF 00 FF 00 FF 00 FF .....
$8078 00 FF 00 FF 00 FF 00 FF .....
+0 +1 +2 +3 +4 +5 +6 +7 01234567
$8080 00 FF 00 FF 00 FF 00 FF .....
$8088 00 FF 00 FF 00 FF 00 FF .....
$8090 00 FF 00 FF 00 FF 00 FF .....
$8098 00 FF 00 FF 00 FF 00 FF .....
--- <Space> or <Return> for more ---
```

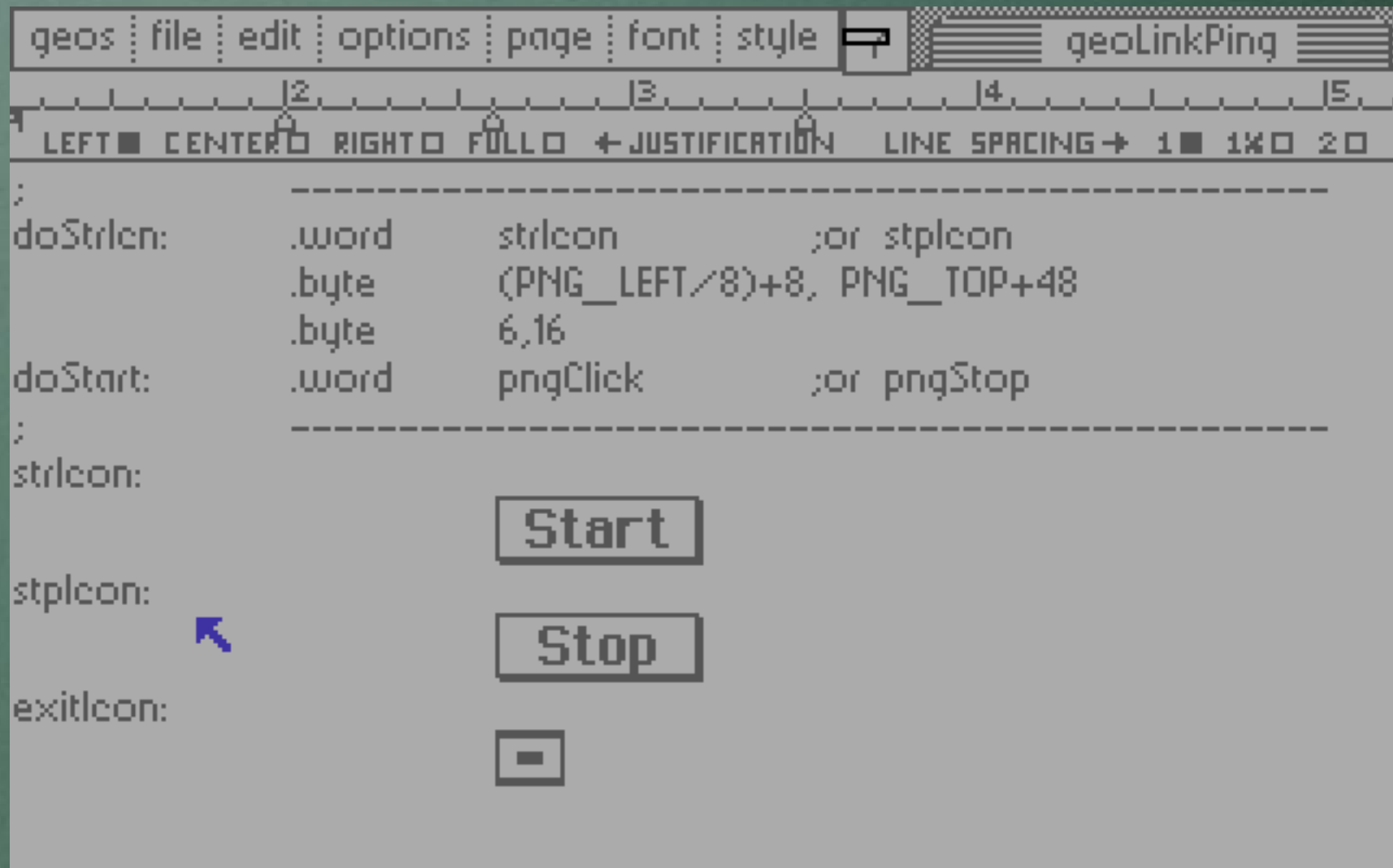
GEOS programming conventions

- ✓ source code written in geoWrite, ability to paste bitmaps into code
- ✓ no traditional assembly listing: use geoDebugger!
- ✓ ZP locations set aside for system and user, named r0 - r15 and a0 - a9 (r0L, r0H...)
- ✓ include file provided for API names and variables
- ✓ macros: LoadW, MoveW, PushW, AddW, CmpW...
- ✓ conditional assembly




header source



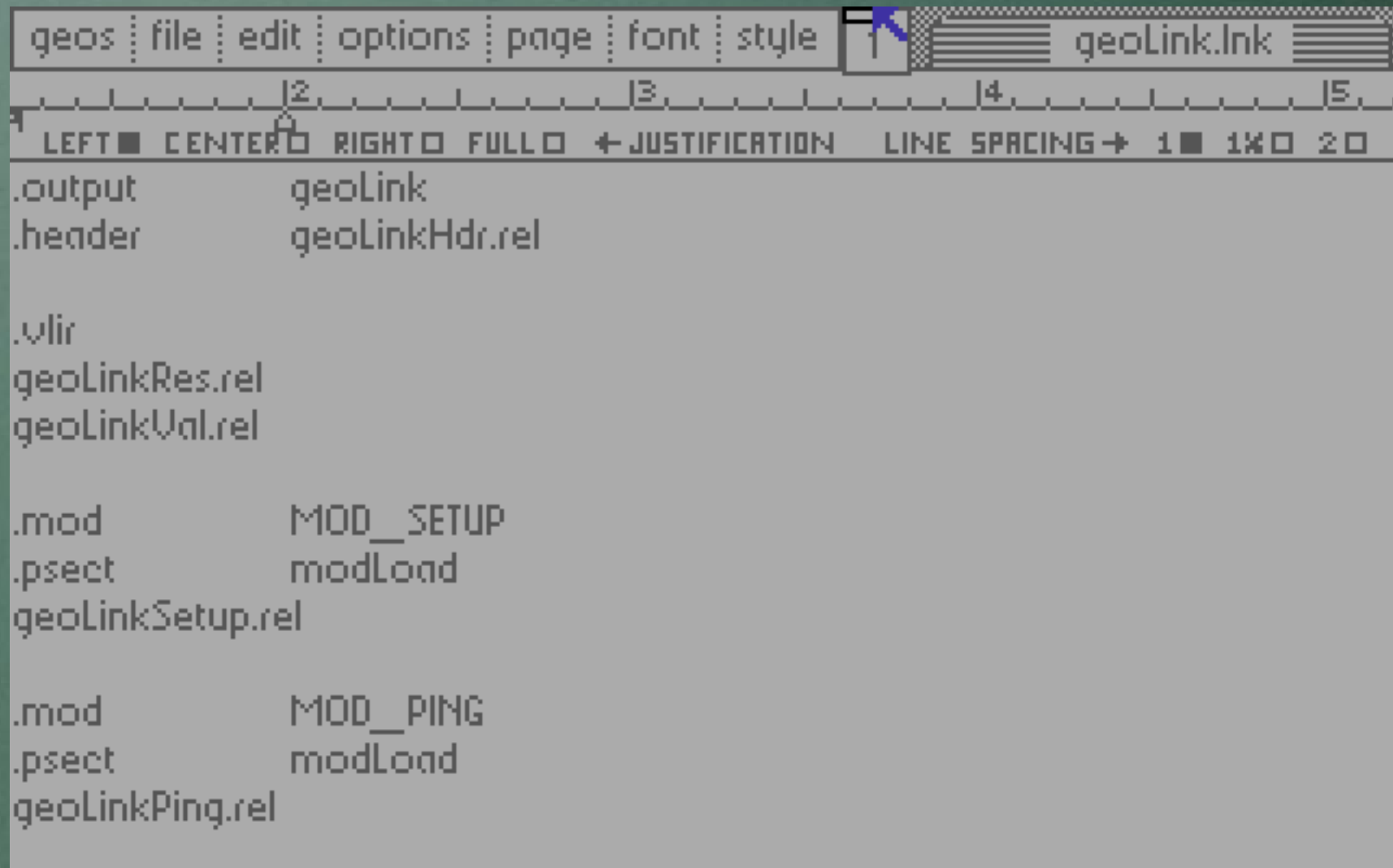
icon bitmaps pasted into code



resident module source

```
geos | file | edit | options | page | font | style |  |  | geolinkRes | 
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 | 5 | | | |
|---|---|---|---|---|---|---|---|
| LEFT | CENTER | RIGHT | FULL | +JUSTIFICATION | LINE SPACING + 1 | 1X | 2X |
start:  lda      #2                      ;50% stipple
        jsr      SetPattern
        LoadB    r2L,0
        LoadB    r2H,199
        LoadW    r3,0
        LoadW    r4,319
        jsr      Rectangle              ;clear screen
        lda      version
        tax
        and      #$f0
        lsr      a
        lsr      a
        lsr      a
        lsr      a
```

linkage directives



GEOs lifecycle

- ✓ deskTop loads program and does a JSR to the init address specified in the file header
- ✓ program's init code sets up callbacks for menus, icons, and processes, then returns
- ✓ GEOs enters MainLoop (event loop)
- ✓ MainLoop invokes callbacks for events flagged during interrupts

GEOS API overview

- ✓ DoMenu, DoIcons, DoDlgBox, InitProcesses, FreezeProcess, BlockProcess
- ✓ Rectangle, DrawLine, BitmapUp, LoadFont
- ✓ FindFTypes, GetFile, DeleteFile, RenameFile, CalcBlocksFree, FollowChain, GetBlock, PutBlock
- ✓ OpenRecordFile, ReadRecord, AppendRecord
- ✓ BMult, DDiv, CopyString, CmpString, MoveData

GEOS memory map

\$BF40

GEOS kernel

\$A000

foreground screen memory

\$8000

disk buffers/driver, printer drivers

\$6000

background screen memory

\$0400

application code

\$0000

pseudo-registers, system variables

geolink memory map

\$6000

ip65 network stack

\$3300

VLIR

font

modules

setup

ping

login

IRC

\$0400

resident module

geolink implementation

ip65 needs a timer and regular polling

- x timer implemented as an interrupt hook, does nothing more than provide a counter
- x polling done with a GEOS process: fired by MainLoop, will not yield control until completion, may miss a scheduled event
- x outgoing packets sent on event handler (i.e. also fired by MainLoop), also will not yield control until completion

incoming IRC message

- ✓ GEOS process calls poll routine every 10 jiffies:
poll routine has to bank in I/O for ip65
- ✓ TCP open specifies recvMsg as callback: banks
GEOS kernel back in (and out again on return)
- ✓ parselin breaks up packet contents into IRC
messages on CR/LF boundaries
- ✓ parseCmd checks tables for known numeric or
text types, sets up dispatch address

transfer of control during polling

process handler calls
ip65 polling routine
(I/O banked in)

ip65 returns to
process handler
(I/O banked out)

polling routine calls
TCP callback
(I/O banked out)

TCP callback
returns to ip65
(I/O banked in)



debugging the IRC client

- ✓ double-click geolink.dbg file to start debugger
- ✓ start IRC client, hit RESTORE to enter debugger
- ✓ set symbol table priority (setmod 4,0,*)
- ✓ set breakpoint where debugging is needed
- ✓ F7 views hi-res screen from command line
- ✓ fix bug
- ✓ lather, rinse, repeat

geoDebugger in action

```
Copyright (C) 1987 Berkeley Softworks  
Program file: geoLink loaded.
```

```
loading symbol definitions.....
```

```
0400 start                > lda #$02  
>a loadirc  
055B loadIRC             ldx  #$00  
055D loadIRC             jsr  FreezePr  
0560 loadIRC             lda  #$0E  
0562 loadIRC             sta  r7H  
0564 loadIRC             lda  #$F4  
0566 loadIRC             sta  r7L  
0568 loadIRC             lda  #$04  
056A loadIRC             sec  
056B loadIRC             jsr  swapMod  
056E loadIRC             jsr  modLoad  
>go  
*** Software Breakpoint ***  
056E loadIRC             jsr  modLoad  
>setmod 4,0,*  
>■
```

geodebugger (no labels)

```
Copyright (C) 1987 Berkeley Softworks  
Program file: geoLink loaded.
```

```
loading symbol definitions.....
```

```
0400 start > lda #$02  
>opt  
opt0 Input radix: (0) hexadecimal  
opt1 Output radix: (0) hexadecimal  
opt2 Labels: (1) disabled  
>a loadirc  
055B A2 00 ldx #$00  
055D 20 12 C1 jsr FreezePr  
0560 A9 0E lda #$0E  
0562 85 11 sta r7H  
0564 A9 F4 lda #$F4  
0566 85 10 sta r7L  
0568 A9 04 lda #$04  
056A 38 sec  
056B 20 C3 06 jsr swapMod  
056E 20 F4 0E b jsr modLoad  
>■
```

a more complex expression

```
>d tcpStrc
+0 +1 +2 +3 +4 +5 +6 +7 01234567
$0C30 52 A5 8B 5F 0B 1A D7 18 R%. .W.
$0C38 15 00 58 52 50 4F 4E 47 ..XRPONG
$0C40 20 3A 69 72 63 2E 74 6F :irc.to

>d ee(tcpStrc+.10)
+0 +1 +2 +3 +4 +5 +6 +7 01234567
$5258 3A 53 68 61 64 6F 77 4D :ShadowM
$5260 21 7E 53 68 61 64 6F 77 !£Shadow
$5268 4D 40 4E 65 77 4E 65 74 MCNewNet
$5270 2D 32 37 33 32 33 44 36 -27323D6
$5278 32 2E 77 65 73 74 2E 62 2.west.b
$5280 69 7A 2E 72 72 2E 63 6F iz.r.r.co
$5288 6D 20 50 52 49 56 4D 53 M PRIUMS
$5290 47 20 23 63 36 34 66 72 G #c64fr
$5298 69 65 6E 64 73 20 3A 74 iends :t
$52A0 65 73 74 20 6D 65 73 73 est Mess
$52A8 61 67 65 0D 0A 53 67 65 age..Sge
$52B0 20 61 74 3A 20 68 74 74 at: htt
$52B8 70 3A 2F 2F 77 77 77 2E p://www.
$52C0 63 36 34 66 72 69 65 6E c64frien
$52C8 64 73 2E 63 6F 6D 2F 0D ds.com/.
$52D0 0A 00 61 6E 20 74 77 6F ..an two
>
```

Resources

geolink 1.01 source code:

<http://lyonlabs.org/commodore/geoLink/geoLink-20100426.d64>

GEOS programming tools and API guides

- x geoProgrammer, geoWrite

- x Programmer's Ref. Guide, Hitchhiker's Guide, Boyce

<http://lyonlabs.org/commodore/onrequest/geos.html#tools>

ip65 networking stack:

<http://ip65.sourceforge.net/>