**See: “How to create DOSBOOT (2022)” on Google Drive**

**1/3/2023:**

* **Getting lost in an infinite series of SCXG1**
* **'27: USERPROG.INITKERN' IPC = 1137**
* **Likely causes:** 
  + **changing OpCode from local variable**
  + **Fetch is setting OpCode (fOpCode) when it did not do so previously**
  + **SetEnableExternalPool is setting Globals.LowMem.SysCom.PoolInfo.pooloutside**
* **OSINIT is calling INIT\_INTERNAL\_POOL**
* **Getting lost in INIT\_TASKS- TASK\_START**
  + **CPL - task\_ start returns to wrong location**
  + **POOLOUTSIDE is false**
* **On the SurfacePro EnableExternalPool is a public variable in INTERP\_COMMON**
* **TIVPsystemInterpreter.PoolBase uses the value of EnableExternalPool**
* **Why is INIT\_INTERNAL being used rather than INIT\_EXTERNAL? PoolOutside is set to TRUE. PoolOutide starts off as FALSE and then gets set to true (on Surface Pro).**
  + **Changed in USERPROG.InitSyscom @ 102 (syscom^.poolinfo := poolinfo;). Coming from SYSTEM.MISCINFO**
* **How to run the version on C:\d7\projects\psystem**
  + **Make sure that the project options all point to the version on the local C: drive**
  + **Make sure that the project manager files all reference local files on the C: drive**
  + **Make sure that R(un Parameters point to the local .ini file (C:\D7\Projects\pSystem\Mountlist.csv)**
  + **Make sure that p-System/Select Version references the correct version**
  + **Check p-System / Enable external pool is set properly**
  + **Start the debugger from filer main**
  + **In the debugger do *Load Version IV / pSysx***

**1/4/2022:**

* **I can successfully boot from PSYSTEMX on SurfacePro but not from Delphi**
* **Trying to boot from PSYSTEMY.VOL gives \*POOL OVERFLOW\* on both SurfacePro and on Delphi. [1] See 6/17/2023 below**
* **Delphi PSYSTEMX gives “System error in ReadSeg” after copying SYSTEM.MISCINFO from the SurfacePro Version**

**1/5/2022:**

* **Where did I put the list of addresses? They are in the “2022 How to Create DOS Boot” on 12/27/2022.**
* **Where are my notes about how to debug PSYSTEM.EXE? They are in “\\xps-8930\ndas-i\d7\Projects\pSystem\Docs\20221223-notes.rtf”**
* **On VPC I run TDX, do ^F9 and then ^F8 to get to the fetch loop. This will boot SYSTEM4Z.VOL (SYS4Z:).**
* **BUBUTIL: PSYSTEM.CFG (20 bytes / record). Appears to be 6 records.** 
  + **Volume number :4, State: OnLine, Volume Class:vfile; Volume Location: C:SYSTEM4Z.VOL, Mode: R/W; VolumeSize 10002**
  + **Volume number :4, State: OnLine, Volume Class:vfile; Volume Location: C:UTILITY.VOL, Mode: R/W; VolumeSize 10000D**
  + **TConfigRec = record**
  + **P: PACKED ARRAY[0..1] of byte;**
  + **Location: PACKED ARRAY[0..13] OF CHAR;**
  + **Stuff: PACKED ARRAY[0..3] OF byte;**
  + **end;**
  + **‘P’ has two bytes: (2,2) before the file name and 4 bytes after**
* **The PSYSTEM.CFG file has records that are 20 bytes long.**
* **To get info BORLAND Pascal 7.0, open the “Development” folder**
* **HpLaptop seems to mung up PSYSYTEM.CFG. You cannot boot without the PSYSTEM.CFG. If it is read locked you get a “Configuration file could not be loaded” message.**
* **What is all the above about?** 
  + **I want to be able to boot a minimal version of SYSTEM4Z.VOL. I.e., no subsidiary volumes and no UTILITY.VOL. Maybe I can just use UEDIT32.EXE to zero out “UTILITY.VOL”.**
  + **I can zap the UTILITY.VOL portion of PSYSTEM.CFG and boot.**
* **SCREENHEIGHT- Cannot change this without a bunch of mucking around**
* **I haven’t succeeded in changing the screen height to 50 lines. See FIXINFO.TEXT in MISCSRC.SVOL on SYSTEM4.VOL**
* **2000 in last word of record of PSYSTEM.CFG??**

**1/6/2023:**

* **SCREENHEIGHT, SCREENWIDTH- changing does not seem to make any difference!**
* **I have lost my notes. “\\xps-8930\ndas-i\d7\Projects\pSystem\Docs\20221223-notes.rtf” is not the latest version. MIght it have gotten lost in the crash? Search for “How to Create DOSBOOT” for old versions.**
* **\\xps-8930\ndas-i\d7\Projects\pSystem\Docs\Turbo\_Debugger\_Version\_5\_Users\_Guide.pdf**
* **STARTING OVER:**
  + **tdx.bat**
  + **^F9**
    - **SET A BREAK AT CS:0E8DH (52C0:0E8D)**
    - **RUN TO THE BREAK POINT**
    - **step into it**
    - **set a break at cs:0f35**
    - **RUN TO IT**
    - **cs:1185 (mov dx,bx) is procedure OPEN\_FILE**
    - **STEP OVER CALL cs:1172**
    - **SET A WATCH ES:DI,S (SegName?)**
    - **SET A BREAK ON CS:1013 (5CC0:1013)**
  + **^F8**
    - **SET A BREAK AT CS:197 (AND RUN TO IT)**
    - **ADD A WATCH OF [CS:0197],M - This will display when the code at cs:0197 has been loaded- 4B 20 ⇒ FF A7 after change. Can I set a breakpoint after the change?**
    - **Add a watch: [cs:00b8] to display the current procedure number**
  + **Delete obsolete breakpoints**
  + **SINGLE STEP FOR A WHILE**
  + **Add the breakpoint at cs:0197 and run to the break**
  + **Try a break on GETPOOL (1119) Code does not look like Delphi?**
  + **CHANGE THE BREAKPOINT AT CS:197 TO HAVE a "LOG" ACTION OF**
  + **LOG "WORD CS:00B8H", ALWAYS - Do not include the brackets!**
  + **IT SHOULD BE SAFE TO SET A BREAKPOINT AT CS:0197**
  + **PROCEDURE SEQUENCE 0, 4, 3, 2, 5, 6, 9, 7, 12, 13, 3, 2, 14, 1, 58, 41, 58, 40, 42,**
* **starts out in procedure 0**
* **Does ES:DI,s represent the segment name? NO.**
* **How can I see the segment name?**
* **Turbo Debugger terminology: Format specifiers**
* **Maybe I can put breakpoints on each of the individual CALLing operators (CPL, CPG, CXG., SCXG, CXI, SCPI, CPI, CXL)**

**1/7/2023:**

* **Delphi procedure sequence: 1, 4, 3, 2, 3 {never went to 5}, 2, 3, 2, 3, 2, 3, 2,**
* **Try to put a break on READSEG**
* **Log the OpCodes for procedure number 2**
* **SI looks a lot like RelIPC (but maybe relative to the segment rather than to the procedure)**

| **VPC** | **VPC-Proc** | **Delphi** | **Delphi Proc** | **Delphi OpCode** | **DbgCnt** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- |
| **144** | **0** | **144** |  | **cpl** | **0** |  |
| **1** | **4** | **1** | **4** | **sldc1** | **1** |  |
| **112** |  | **112** |  | **scxg1** | **2** |  |
| **134** |  | **134** |  | **lao** | **3** |  |
| **2** |  | **2** |  | **sldc2** | **4** |  |
| **225** |  | **225** |  | **ngi** | **5** |  |
| **157** |  | **157** |  | **lpr** | **6** |  |
| **196** |  | **196** |  | **sto** | **7** |  |
| **133** |  | **133** |  | **ldo** | **8** |  |
| **231** |  | **231** |  | **inc** | **9** |  |
| **1** |  | **1** |  | **sldc 1** | **10** |  |
| **238** |  | **238** |  | **deci** | **11** |  |
| **215** |  | **215** |  | **ixa** | **12** |  |
| **120** |  | **120** |  | **sind0** | **13** |  |
| **239** |  | **239** |  | **scip1** | **14** |  |
| **47** | **3** | **47** |  | **sldl16** | **15** |  |
| **122** |  | **122** |  | **sind2** | **16** |  |
| **164** |  | **164** | **3** | **stl** | **17** |  |
| **0** |  | **0** | **3** | **sldc0** | **18** |  |
| **46** |  | **46** |  | **sldl15** | **19** |  |
| **231** |  | **231** |  | **inc** | **20** |  |
| **6** |  | **6** |  | **sldc6** | **21** |  |
| **239** |  | **239** |  | **scip1** | **22** |  |
| **97** | **2** | **97** | **2** | **slla2** | **23** |  |
| **35** |  | **35** |  | **sldl4** | **24** |  |
| **197** |  | **197** |  | **mov** | **25** | **I am moving bytecount bytes. They are moving words.** |
| **96** | **5?** | **96** |  | **slla1** | **26** |  |
| **33** |  | **33** |  | **sldl2** | **27** |  |
| **34** |  | **34** |  | **sldl3** | **28** |  |
| **215** |  | **215** |  | **ixa** | **29** |  |
| **196** |  | **196** |  | **sto** | **30** |  |
| **32** |  | **32** |  | **sldl1** | **31** |  |
| **108** |  | **108** |  | **sstl5** | **32** |  |
| **150** |  | **150** |  | **rpu** | **33** | **The Delphi rpu returned to proc 3 but vpc didn’t? DbgCnt =34.1** |
| **164** | **2** | **164** | **2** | **stl** | **34** |  |
| **132** |  | **132** |  | **lla** | **35** |  |
| **46** |  | **46** |  | **sldl15** | **36** |  |
| **120** |  | **120** |  | **sind0** | **37** |  |
| **42** |  | **42** |  | **sldl11** | **38** |  |
| **2** |  | **2** |  | **sldc2** | **39** |  |
| **148** |  | **148** |  | **cxg 24** | **40** | **call to GETPOOL** |
| **45** |  | **45** |  | **sldl14** | **41** |  |
| **1** |  | **1** |  | **sldc1** | **42** |  |
| **177** |  | **177** |  | **neqi** | **43** |  |
| **213** |  | **213** |  | **fjpl 446** | **44** | **446 IS THE EXIT {here}** |
| **150** |  | **150** |  | **sldl16** | **45** | **off we go into never never land** |
| **134** |  | **134** |  | **(sldc 7)** | **46** |  |
| **1** |  | **1** |  | **sldc1** | **47** |  |
| **225** |  | **225** |  | **cxg** | **48** | **{here}** |
| **…** |  |  |  |  |  |  |
| **135** |  | **135** |  | **ldl** | **242** |  |
| **0** |  | **0** |  | **sldc 0** | **243** |  |
| **215** |  |  |  | **ixa** | **244** |  |

* **1Maybe what I think is the current procedure number (cs:00b8h) really isn’t the current procedure?**
* **VPC compares the target CSP number against 23 (possibly \*2 = 46 = highest usable CSP opcode)**
* **How can I find the address of GETPOOL? It should be in my jump table addresses.**
* **The CSP jump table starts at $168C**
* **The jump table starts at $1488**

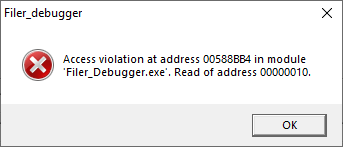
**1/9/2023:**

* **PSYSTEMX says:**
  + **POOLOUTSIDE = TRUE**
  + **POOLSIZE = $0040 [64]**
  + **RESOLUTION = 16**

**1/10/2023L**

* **Version I.4: Error = 129, EnterIC = $8F8C (36748), Lexlevel = 0, S#0, P333, O#129 (Unimplemented OpCode = 212 when I try to H(alt the system).**
* **For some reason SYSTEM.STARTUP does not show up in the directory that I load when guessing? But it does appear if I just mount the volume. It wasn’t mounting the TAppleVolume when trying to guess.**

**1/12/2023:**

* **Getting “Output File” when I try to boot.**
* **Getting “Output File (File (\*.CSV)” when I try to set the Debug Database to use. Was occurring because fLogFileName was empty.**
* **Getting AV in “Load Version” (after entering the unit number)**
* **“VolumesToOpen.csv” is not updated.**
* **Getting an ASSERT error in READSEG: Assert(LocalVar = Globals.Lowmem.MPPLUS, ERRMSG)**

**1/13/2013:**

* **(For PeterW) Look for SetPtr (assembly language).**

**1/14/2023:**

* **Getting the ASSERT error when trying to boot PSYSTEMX (even using the PSYSTEMX from surface pro which formerly worked!) Works OK if I enable External memory.**
* **FlipSeg, DbgCnt = 48**
* **DbgDnt gets to 362 but the breakpoint in CALLIO never trips?**
* **The break occurs at DbgCnt = 362**
* **The DECODE that I did of SYSTEM.PASCAL only includes GETCMD? Did it again and the result was OK.**
* **The call to UNITREAD(0) (HALT) in USERPROG.procedure #6 does not have any jumps to skip over the HALT and there is a heap of code following it?**
* **Maybe UNITREAD(0) is not a HALT in version 4.12?**
* **Preceded by “LDCI 2028” which appears to be SIZEOF(TDirectory).**
* **SCXG 18**
* **Is it possible that the CSPOPS for version IV.12 don’t include all of the ops for the original version?**
* **CSPEND not properly set for VersionIV\_12**
* **Getting a memory leak.**
* **Gets hung during boot**

**1/16/2023:**

* **pCodeDecoderUnit is initializing all of the CSP operators which is not appropriate for Version 4.12**
* **CSP calls:**
  + **34 UCLEAR? This goes into an infinite loop of seg faults.**

**1/17/2023:**

* **Getting into SurfacePro TD debugger**
  + **Boot Surface Pro**
  + **Oracle VM VirtualBox**
  + **Boot Windows XP**
  + **pays**
  + **run “cmd”**
  + **TDX**
  + **On VPC I run TDX, do ^F9 and then ^F8 to get to the fetch loop. This will boot SYSTEM4Z.VOL (SYS4Z:).**
* **TESTING SCXG on VPC: Put a breakpoint at 0d44 and 0CF9**
  + **Break on SCXG**
    - **The CSP procedure number is compared to $027 (39D). Its value**
    - **CXG procedure on VPC:**
      * **$022 (18D UREAD)**
      * **$018 (24D GETPOOL)**
      * **$00f (15D MOVELEFT)**
      * **$012 (18D UREAD)**
      * **$012 (18D UREAD)**
      * **$00F (15D MOVELEFT)**
      * **$012 (18D UREAD)**
    - **CXG numbers on Delphi:**
      * **34D (UCLEAR)**
      * **24D (GETPOOL)**
      * **26D (FLIPSEG)**
      * **24D (GETPOOL)**
      * **26D (FLIPSEG)**
      * **24D (GETPOOL)...**
      * **18D (UREAD) of 0-- HALT**
    - **If greater (or equal) than 39D (READSEG)**
* **I’m still getting a CallIO on unit 0. I.e., HALT**
* **What calls GETPOOL? Setting a breakpoint on VPC at $0119 never gets hit. Delphi hits GETPOOL at DbgCnt = 40.**
* **Does the MOV op at DbgCnt 25 change the CurProc?**
* **REPNZ MOVSW: Moves CX words from DS:[SI] to ES:[DI]**
* **I really need to display the DbgCnt.**
  + **To do that, I need to find a usable section of memory for the patch.**
  + **See *PME-Debug.txt***

**1/18/2023:**

* **CHK is located at 0705**
* **That should allow me to see what happens when there is an invalid index (01F0)**
* **INVNDX is located at 01F0**
* **Which jumps to 01D2 which is XXEQERROR**
* **I need an unused location to store DbgCnt** 
  + **I could use the low end of the stack $0203**
  + **I could overwrite the constant pool $54E**
* **Here is the patch:**

**CS:01D2 a13a00 mov ax,[003a] mov ax,DbgCnt**

**cs:01d5 40 inc ax inc ax**

**cs:01d6 a33a00 mov [003a],ax mov DbgCnt,ax**

**cs:01d9 32ff xor bh,bh**

**ebb3 jmp 191 jmp fetch+**

**Here is the patch for cs:018f:**

**cs:018f eb41 jmp cs:01d2**

**DbgCnt located at cs:[003a]**

**I need to have PMACHINE.IV1 in the boot directory. This is the interpreter.**

* **FETCH is at 18F**
* **There needs to be code at 018F**

**1/19/2023:**

* **RPU is located at $0BCF**
* **GETBIG returns word count to BL**
* **NEWENV sets SEGB (no where else)**
* **Is GetPool expected to restore it?**
* **Putting s break on GXG (148) which is the call to GETPOOL**
* **Trying to execute CSP procedure 24**
* **CS:11f9 is GETPOOL**
* **Look for ES getting changed**
* **Possibility: GetPool needs to be restoring some register**
* **Something is mangling DbgCnt when I put a breakpoint on ES**
* **ES:0 is the base of the Segment**
* **Files that I need to have open:**
  + **How to create DOSBOOT (2022) {Google Docs}**
  + **How to create DOSBOOT (2023) {Google Docs}**
  + **F:\NDAS-I\d7\Projects\pSystem\AccDb\VersionIV.accdb**
  + **Decode of System.pascal from SYSTEM4-12-VOL.TXT (Delphi)**
  + **Interp4.pas (Delphi)**
  + **Oracle / TDX.BAT / C:\PSYS\PSYSTEM0.EXE System4x.vol**
  + **Jump Table for System4 (OUTPUT).xls**
  + **Turbo\_Debugger\_Version\_5\_Users\_Guide.pdf**
  + **8086\_Instruction\_set.pdf**

**1/20/2023:**

* **Does the call to FLIPSEG execute p-Code or internal interpreter code?**
* **Further setups to do in the TD debugger:**
  + **Watch bx / 2 (i.e., OpCode)**
  + **Set a break @ cs:0197 (this is in FETCH)**
  + **Set a break condition on that break of**
    - **(bx div 2) = some opcode {the language will have to be set to Pascal to do this)**
  + **The segment is loaded at $DDC0. Byte sex is at $DDCC but the referenced address (bx) is at $DDA2 = 2A (42D) bytes away 🙁**
* **Important watches:**
  + **B8 = Procnum**
  + **(bx / 2) or (bx div 2) = Opcode**
  + **3A = DbgCnt**
* **Important Breakpoint**
  + **cs:0197 (in FETCH just before ProcCall)**
  + **CS:2333 (in UREAD) {here}**
* **Picking up a local variable at offset 14. Value loaded is 0.**

**1/23/2022:**

* **With debugger**
  + **Creation Order**
    - **TfrmpSysWindow (from TfrmFiler)**
    - **TIVPsystemInterpreter (from TfrmFiler)**
    - **TfrmPCodeDebugger (from TfrmFiler)**
  + **Destruction Order**
    - **TfrmPCodeDebugger**
    - **TIVPSystemInterpreter**
    - **TfrmPSysWindow**
* **Filer Only**
  + **Creation Order**
    - **TfrmPsysWindow (from TFrmFile)**
    - **TIVPsystemInterpreter (from TfrmFiler)**
  + **DestructionOrder**
    - **TIVPsystemInterpreter**
    - **TfrmPSysWindow**
  + **Destruction order when manually closing the p-Sys window**
    - **TfrmPSysWindow**
    - **(closing the Filer Window manually)**
    - **TIVPsystemInterpreter**
* **Getting integer overflow in PUSH?**
* **Why isn’t “System Halt” centered on the main form?**
* **Any messages that are attempted during the closing down process are lost forever 🙁 See:** <https://stackoverflow.com/questions/28270559/why-is-messagedlg-not-shown-after-a-form-is-freed-via-onclose-event-in-a-thread>
* **Why doesn’t the BootItems list include the “Specify” item? By the time that SaveToFile is called, there are only two items in the list. The IsClean still had a reference to FileExists(DebugDBToUse) even in the non debugging filer.**
* **When I try to re-boot without restarting, things seem hung. If I then quit, I get memory leaks.** 
  + **I think that maybe a new interpreter was created without freeing the old one. (The interpreter is destroyed when the file is destroyed)**
  + **Now if I restart, the MISCINFO does not get reset-- even if I try to force VT52.**
* **GoToXY is not recognized when trying to boot the 2nd time. Maybe the GoToXY prefix is not recognized. INDEX[] never got set -- none of them!**
  + **CRTINFO.SetTermType never got a chance to SetTermType. fTermType is already set to tt\_VT52**

**1/24/2023:**

* **“Enabled External Pool”- default to TRUE (this setting has been deleted)**

**1/25/2023:**

* **What is cbAccDbFileName used for in BreakPointInfo? To know which DB to use when setting the breakpoint segment name, procedure name**
* **The pnlParam should be visible for the following kinds of breakpoints: dbBreak, dbDbgCnt, dbOpCode**
* **Have I screwed up Watch points? No.**
* **What does dbDecode, dbBreakOnCall, dbSYSTEM\_HALT, do? They are used to break out of the FETCH loop. The user cannot set them.**
* **dbMemoryChanged isn’t displaying needed parameters.**
* **Breakpoint at 18503 during a READSEG operation.**
  + **Happens during call to procedure LOAD on a READSEG**
  + **Call to POOLBASE which returns 0. What is the poolbase when there is no external pool? POOLBASE has already been called many (35+) times returning a 0 value.**
  + **This set ES to 0 for the first time? This is the first time that PoolBase has been called with a non-zero PoolDescInfoAddr value**
  + **PoolBase(Seg\_Pool) + SegAddr**
* **CONCLUSION: The problem with PoolBase is that it is setting ES to zero when it should not be doing so!** 
  + **ES := PoolBase(Seg\_Pool)**
* **Does PSYSTEM.EXE have the setting for extended memory turned on?**

**1/26/2023:**

* **PSYSTEM.EXE does have a setting (SETUP.CODE): (none of these are changed by the interpreter)**
  + **“CODE POOL BASE[FIRST WORD]” with a value of 1 (i.e., $10000).**
    - **Change shows up in SYSTEM.MISCINFO** 
      * **35: 7F->6F**
      * **42: F9->A0 (timestamp)**
      * **43: 52->D9 (timestamp)**
  + **“CODE POOL BASE[SECOND WORD]” with a value of 21696 (i.e., $54C0->$54B0).**
    - **38: c0->b0**
    - **42: f9->59 (timestamp)**
    - **43: 52->ec (timestamp)**
  + **“CODE POOL SIZE” with a value of 32767 (I.E., $7FFF->$6FFF)**
    - **35: 7f->6f**
    - **42: F9->82 (timestamp)**
    - **43: 52->32 (timestamp)**
  + **“HAS EXTENDED MEMORY” value is TRUE->FALSE**
    - **32: 01->00 (pooloutside)**
    - **42: F9->BD (timestamp)**
    - **43: 52->3D (timestamp)**
  + **“SEGMENT ALIGNMENT” value is 16 ($10->$20)**
    - **40: 10->20 (resolution)**
    - **42: F9->96 (timestamp)**
    - **43: 52->48 (timestamp)**
* **Does my code now use INIT\_EXTERNAL\_POOL rather than INIT\_INTERNAL\_POOL? ~~. Always calling INIT\_EXTERNAL\_POOL.**
* **USERPROG.ALLOCATE loads *pooloutside* from SLDL 3; SIND0**
* **All of the code for SetEnableExternalPool is probably irrelevant and needs to be rethought. When the OS loads SYSTEM.MISCINFO it's going to be overwritten!**

**1/27/2023:**

* **I moved BLOCKSIZE & TBlock into Interp\_Decl**
* **Their version of PoolBase returns DS when PoolDescInfoAddr is 0.**
* **PoolBase(Seg\_Pool) + SegAddr**
* **Code Pool Info should allow HEX data entry and should do more error checking. Entering 0 for codepoolbase when pooloutside was true caused crashes.**

**1/30/2022:**

* **Version IV call to PoolBase never returns 0 in READSEG**
* **PSYSTEMY.VOL (“\\hplaptop\PSYS\psystemy.vol”) (“Power System”?) on HPLAPTOP returns 0 several times but boots anyway.**
  + **PoolOutside = false**
  + **PoolSize = $7FFF [32767]**
  + **PoolBase = 00000000**
  + **Resolution = 16**
* **F:\NDAS-I\d7\Projects\pSystem\Volumes\PSYSTEMX.VOL never returns 0 in ReadSeg / PoolBase**
  + **PoolOutside = true**
  + **PoolSize = $003F [ 63]**
  + **PoolBase = 0001FC00**
  + **Resolution = 16**
* **Search PoolInfo notes:**
  + **Not displaying “SYSTEM.MISCINFO”**
  + **No Default output file name**
  + **When PoolOutSide is false, the PoolSize is $7BFF or $7FFF and the PoolBaseAddr if $10000+**
  + **When PoolOutside is true, PoolSize might be: $3F, of $0100 or $7FFF**
  + **Resolution is ALWAYS 16**

**1/31/2023:**

* **Search for LOADFILES.TEXT, SETUP.CODE**
* **The SYSTEM4 (SYS4:, SYSTEM4:) volumes all have a** 
  + **PoolBase[0] = 1**
  + **PoolBase[1] = $54C0+ (or $5410)**
  + **PoolSize = $7FFF**
  + **PoolOutSide = TRUE**
  + **Resolution = 16**
* **The PSYSTEMx (SYSTEM:) volumes all have:**
  + **PoolBase[0] = 1**
  + **PoolBase[1] = 0**
  + **PoolSize = $7BFF**
  + **PoolOutSide = TRUE or FALSE**
  + **Resolution = 16**
* **When I try to execute SETUP.CODE from a SYS4: (SYSTEM4.VOL), the POOLINFO values seem to be mixed up and incorrect. Other values such as CRT/Keyboard info seem to be correct. Seems like maybe the location for POOLINFO got changed between version 4.12 and version 4.2? e.g.: (this might be due to Syscom having been moved.)**
  + **CODE POOL BASE[FIRST WORD]: 09**
  + **CODE POOL BASE[SECOND WORD]: CB 30**
  + **CODE POOL SIZE: 020E**
  + **None of the above values show up in SYSTEM.MISCINFO when copied to DOS?**
  + **When I examine SYSTEM.MISCINFO using UEDIT32.EXE (or editing the entire SYSTEM.VOL) all the values (including CRT/KEY info) look fine. I don’t know where SETUP.EXE is getting its POOLINFO values from.**
* **PSYSTEM.COM expects to boot from PSYSTEM.VOL by default**
* **I am changing the POOLINFO settings on F:\NDAS-I\d7\Projects\pSystem\Volumes\SYSTEM4.VOL to have a ZERO value for POOLBASE[0] AND POOLBASE[1]**
  + **I am no longer getting the SYSTEM HALT in READSEG because of ES = 0. Dying in (ReadSegRefs (was FILEPRES) supposedly) DbgCnt = 363. This is EXACTLY where things died before. Changing POOLBASE didn’t fix anything.**

**2/1/2023:**

**Q. What is the problem?**

**A: When booting Version 4.12 (SYSTEM4.VOL), the system always HALTs very quickly after booting.**

**Q. What have I learned about this problem?**

1. **Preceded by “LDCI 2028” which appears to be SIZEOF(TDirectory). This occurs at DBGCNT = 360.**
2. **CSPOPS for version IV.12 does not include all of the ops for the original version 4.2?**
3. **May have something to do with GETPOOL**
4. **Since setting PoolBase to 0,0 (with Pool OutSide=true and PoolSize = $7FFF) I do not get a break in READSEG following the call to PoolBase(Seg\_Pool)**
5. **Version 4.2 sets ES to 0 many times. Implication is that the segment is in low memory (even though “Pool Outside” is true).**
6. **PoolSize (sometimes?) represents the number of “chunks”?**
7. **The VPC version pf 4.12 and the Delphi version diverge when VPC finds something (?) = 1 but Delphi does not. This occurs around DbgCnt = 45.**
8. **If poolsize in MISCINFO is 0, then allocate structures should fill it in.**
9. **There is some indication that poolsize is in k bytes (at least when pooloutside = true).**
10. **There is a Filer SET (version 4.2) command which will display settings such as POOLSIZE. The SET command does not exist in Version 4.12**
11. **SYSTEM4 (4.12 usually specifies poolsize in SYSTEM.MISCINFO)**

* **At DbgCnt = ~40 I am calling GETPOOL. Is VPC calling something else? It is located at $11F9; this is for CSP Op 24 (which is probably GETPOOL). This GETPOOL returns with the same address as the code segment (CS) and the stack segment (SS) and the data segment (DS). ($523c)**

**2/2/2023:**

* **Trying to get a SEGMAP of SYSTEM.PASCAL from SYSTEM4 leads to AV (sometimes-- possibly needed to restart).**
* **I think that the code IS actually FLIPSEG and that it is trying to examine the SEX field within the segment info. Try putting a break in after READSEG.**
* **UserBase is $E7A0**
* **SYSTEM.PASCAL starts at block 6. USERPROG starts at 58. 58+6 = 64 which is where it is reading from.**
* **SYSTEM.PASCAL (SYSTEM4) actually has 27 segments but my SEGMAP only shows the first 16 of them.**
* **The data that I need to display starts at $E7A0**
* **VarType() = 19 isn’t printing anything. This was because the type was vtLongWord and AddPiece was not handling it.**

**2/3/2023:**

* **Why is Map[1] = 0?**
* **The initial EVECp is @ $E76C**
* **'EVEC @ $E76C [59244]: Vect\_Length=2, Map[ 1]=0000, Map[ 2]=E772'**
* **On VPC, the call to LPR ought to show me where the EVECp is and I can look at the Map[0] to see what its value is.**
* **Their EVEC @CS:00c4 seems to be fully loaded by the time that LPR gets called. It seems to be loaded before the first p-Code instruction gets loaded.**
* **The LOOP instruction does not get traced in the debugger**
* **I think that a procedure table from the segment is getting adjusted and put into the EVEC Map[n] fields?**
* **I am currently working on a *Search* procedure to scan code files and report on the procedure offsets for each procedure.**
* **I do not see any segments with 16 procedures?**
* **Also the vector (?) seems to imply $0020 = 32 procedures?**
* **This is NOT a procedure vector. It is an environment vector. Each entry in the vector points to a combo EREC immediately followed by a SIB.**

**2/6/2023:**

* **When setting a changed memory global on cs:00c4 (the assumed vector table), the code that makes the change is located cs:3909**
* **If this is an EVEC, it does not make any sense. Each of the entries in the table differs by $2E (=46).**
* **What does VPC do on a UNITREAD to unit 0?**
* **Could this be doing something with the semaphores?**
* **UNITREAD (vpc) is located at $2333**
* **UNITWRITE (vpc) IS located at $2338**
* **The UNUM may be accessed at $2240**
* **I may be screwing something up by assuming that location CS:003A is not being used for something else.**
* **Putting a breakpoint at 233A should catch both UNITREAD & UNITWRITE**

**2/7/2023:**

* **It appears as if I made my patches via TD macros rather than patching the exe files**
* **This EVEC is located at $cs:00c4**

**ds:00C4 0020 FFD0 FFA2 FF74**

**ds:00CC FF46 FF18 FEEA FEBC**

**ds:00D4 FE8E FE60 FE32 FE04**

**ds:00DC FDD6 FDA8 FD7A FD4C**

**ds:00E4 FD1E 0000 0000 0000**

**ds:00EC 0000 0000 0000 0000**

**ds:00F4 0000 0000 0000 0000**

**ds:00FC 0000 0000 0000 0000**

* **In the EVEC are what the addresses refer to (see \\xps-8930\ndas-i\d7\Projects\pSystem\TEMP\FIBDUMP.LOG :**
  + **FFD0 KERNEL**
  + **FFA2 GETCMD**
  + **FF74 INITOPS**
  + **FF46 HEAPOPS**
  + **FF18 EXTRAHEA**
  + **FEEA PERMHEAP**
  + **FEBC COMMANDI**
  + **FE8E SMALLCOM**
  + **FE60 SOFTOPS**
  + **FE32 STRINGOP**
  + **FE04 PRINTERR**
  + **FDD6 OSUTIL**
  + **FDA8 CONCURRE**
  + **FD7A PASCALIO**
  + **FD4C EXTRAIO**
  + **FD1E USERPROG**
* **Each one of these addresses points to a block of 46 bytes.**
* **Each block contains an EREC of 10 bytes immediately followed by a SIB of 36 bytes.**

**Dump**

**ds:FFD0 0C EC C4 00 DA FF 01 00 ìÄ Úÿ EREC of 10 bytes**

**ds:FFD8 00 00 00 00 10 EE 01 00 î SIB of 36 bytes.**

**ds:FFE0 00 00 01 00 FF FF 4B 45 ÿÿKE**

**ds:FFE8 52 4E 45 4C 20 20 81 07 RNEL**

**ds:FFF0 07 00 04 00 00 00 00 00**

**ds:FFF8 00 00 00 00 00 00**

**Dump 0000 ENV\_DATA**

**ds:FFA2 00 00 C4 00 AC FF 01 00 Ä ¬ÿ 00C4 ENV\_VECT**

**ds:FFAA 00 00 00 00 00 00 00 00 FFAC ENV\_SIB**

**ds:FFB2 00 00 01 00 00 00 47 45 GE 0001 LINK\_COUNT**

**ds:FFBA 54 43 4D 44 20 20 53 0F TCMD S 0000 NEXT\_REC**

**ds:FFC2 0F 00 04 00 00 00 00 00**

**ds:FFCA 00 00 00 00 00 00**

**Dump**

**ds:FF74 00 00 C4 00 7E FF 01 00 Ä ~ÿ 0000 SEG\_POOL**

**ds:FF7C 00 00 00 00 00 00 00 00 00C4 SEG\_BASE**

**ds:FF84 00 00 01 00 00 00 49 4E IN 0000 SEG\_REFS**

**ds:FF8C 49 54 4F 50 53 20 66 06 ITOPS f 0000 TIMESTAMP**

**ds:FF94 1F 00 04 00 00 00 00 00 0001 SEG\_PIECES**

**ds:FF9C 00 00 00 00 00 00 0000 RESIDENCY**

**INITOPS SEG\_NAME**

**0666 SEG\_LEN**

**001F SEG\_ADDR**

**0004 VOL\_INFO {HERE}**

* **UJP +274 did not work? Or the disassembler got it wrong.**
* **Should show as UJP 18**
* **I am making it as far as DbgCnt = 17578. Getting an “odd address” error in SETWORD (UNITREAD on unit 0 ignored many times)**
* **I need to figure out what UNITREAD(0) is trying to do.**

**2/9/2023:**

* **Fix UJP in the disassembler. First call to UJP is supposedly in USERPROG.FLIPSEG.**
* **Does VPC call READSEG before the first p-Code is executed? On VPC, ReadSeg is located at $130D. No. VPC never calls READSEG.**
* **UREAD is located at 2333.**
  + **a breakpoint at $235D should show the**
  + **UNITNUMBER on the TOS**
  + **UBLK @21Ba**
  + **ULEN @21B8**
  + **UBUF @21B6**

| **Read #** | **BLK** | **Len** | **BUF** | **Unit#** | **DbgCnt** |
| --- | --- | --- | --- | --- | --- |
|  | **21BA** | **21B8** | **21B6** | **TOS** |  |
| **1** | **0** | **0** | **0** | **1** |  |
| **2** | **0** | **0** | **0** | **4** |  |
| **3** | **2 (directory)** | **2048 (= 4 blocks)** | **$3ab6** | **4** |  |
| **4** | **6 (system.pascal, segment dict)** | **512 (= 1 block)** | **$3ab6** | **4** |  |
| **5** | **7 (system.pascal, seg0 = kernel)** | **3842** | **$ee10** | **4** |  |
| **6** | **64 (system.pascal, userprog?)** | **3652** | **$ddc0** | **4** |  |
| **7** | **0** | **3652** | **$ddc0** | **1?** |  |
| **8** | **2 (directory)** | **2028 (directory)** | **$3aa8** | **4** | **236** |
| **9** | **1057 (system.miscinfo)** | **96** | **$dd3c** | **4** | **346** |
| **10** | **6 (system.pascal, segment dict)** | **512** | **$d064** | **4** | **2526** |
| **11** | **136 (last block of SYSTEM.PASCAL?)** | **512** | **$D264** | **4** | **3355** |
| **12** | **14 (segment dict in SYSTEM.PASCAL)** | **512** | **$Db3C** | **4** | **18085** |
| **13** | **30 (segment dict in SYSTEM.PASCAL)** | **512** | **$DB3C** | **4** |  |
| **14** | **37 (ditto)** | **512** | **$db3c** | **4** | **27148** |

* **UREAD does not get called before the first p-Code is executed. There must be some other function which is loading SYSTEM.PASCAL**
* **Trying a breakpoint at 2238 which is some function called by UREAD.**
* **To create initial EVECs, ERECs, and SIBs, I would need to load SYSTEM.PASCAL and create all of the stuff on the stack.**
* **There is no evidence of a UNITREAD(0)**
* **On Delphi, the CallIO specifying unit number 0 occurs at DbgCnt 363. I cannot directly compare the Delphi/VPC DbgCnt values because the the systems diverged around DbgCnt 45. Why did they diverge?**
* **When does Map[1] get used/changed?**

**2/10/2023:**

* **Globals.LowMem.EVECp = $E760**
* **EVECAddr = $E760**
* **ERECAddr = $E756, $E70E**
* **SIBAddr = $E732, $E704**
* **Has MSCW address been set? I need to set GLOBALS.LOWMEM.ERECp and EVECp and SIBp. Which one do I use?**
* **I am getting an ODD address error in SetWordAt (@ DbgCnt 17364). This is occurring in USERPROG.LOAD @ 161.**
* **unitread(0) gets passed: UREQ=1, UCTL=0, UBLK=0, ULEN=512, UBUF=$D6F6. There is nothing in the buffer.**

**2/13/2023:**

**STO ->**

**Map[1]: $06c4 → $E74A**

**Map[2]: $06C4 → $06C4**

**Map[3]: $06C4 → $5461 (crash)**

**Loading from an intermediate address via call SLOD1 1381. Intermediate address in (presumably “USERPROG.ReadSegRefs (was FILEPRES)”)**

* **The goal is to find out where these numbers (on the right) came from**
* **Is there a SIB at $E74A and is it wacky?**

**2/14/2023:**

* **The numbers on the right (above) appear to be calculated addresses since they cannot be found in memory.**
* **SLOD1 1381 is referencing a local variable in ReadSegRefs (was FILEPRES). The value stored in $1381 is $06C4.**
* **Can I put a breakpoint on VPC and look for the UNITREAD near DbgCnt = 234? It somehow seems like VPC is magically doing a UNITREAD on unit 4 where as I am doing a UNITREAD on 0**
* **Is location 1380 supposed to contain a unit number? Perhaps from Vol\_Info field from the SIB? Looks like the VOL\_INFO field is not getting initialized.**

**2/15/2023:**

* **The SIB @ $E732 is OK when exiting *Load\_Psystem***
* **Somewhere the VolInfo (4) gets changed to a pointer to a TVip. This is occurring in *load* IPC=160.**
* **Put a memory changed break on $E74A**
* **The SetWordAt odd address error now occurs in USERPROG.LOAD, ipc = 2669, DbgCnt = 2669 (or is it $2635). SegNr = 1**
* **The STOres goto: SegNr=0→$E0F8, SegNr=1→$E70E**
* **$1283 must be a pointer to some sort of array with 16 byte elements-- maybe something in the segment dictionary? No.**
* **I’m hitting the LOAD breakpoint before the ReadSegRefs (was FILEPRES) breakpoint which should not be happening.**
* **UNITREAD: UNUM=4, UBLK=2, ULEN=2028, UBUF=1732. Reads the directory but not to where GDIRP points.**
* **WordAt[$1283] = $E0F8 which points to zeroed memory**

**2/16/2023:**

* **Look at 1380 in “ReadSegRefs (was FILEPRES)”**
* **Copying from $E74A → $E1BA**
* **$6C4 is where the directory gets loaded**
* **Loads 96 bytes of SYSTEM.MISCINFO to $E3CE. This gives you the entire SYSTEM.MISCINFO. Other than CRTINFO, it seems to be garbage.**
* **PROCEDURE UNITREADC UNITNUMBER : INTEGER;**

**VAR DATAAREA : PACKED ARRAY [0..BYTESTOTRANSFER-1] OF 0..255;**

**BYTESTOTRANSFER : INTEGER**

**[; LOGICALBLOCK :INTEGER]**

**[; CONTROL : INTEGER] );**

* **Getting a “Range Check Error” (maybe when LocalVariables gets repainted)? This occurs when I return to the debugger from doing something else. It always shows up in DefaultKeyInfo (which should not even be called) in the FormPaint call stack.**

**2/17/2022:**

* **The local variables for OS\_INIT start at $E442**
* **On the displayed call stack, the IPC, ENV, and PROCNR fields are taken from the previous MSCW**
* **I calculate that var1283 should be located at $E0F8 but the debugger shows it at $E0F0? $E0F8 is what gets loaded as an address.**
* **The addresses being stored to increase by 1 word each time through the loop but it's always a 0 that gets stored**

**2/18/2023:**

* **LOCALVAR should be $D6F2 for ReadSegRefs (was FILEPRES). But when displaying it via intermediate variables, it is using $D6EA which is 8 bytes too low.**
* **Look for references to 1283 in the decoded records in the DB. It is contained in:**
  + **ReadSegRefs (was FILEPRES) (LLA) @ 341, 873, 1015, 1035, 1261, 1293, 1383, 1408, 1423, 1439**
  + **LOAD (LDAn) @ 23, 161, 356**
* **In LOAD at 161, ReadSegRefs (was FILEPRES) at 1283 the value is $E756. If taken as an address it points to the following data: Bytes @ $E756 [59222]: 70 01 60 E7 32 E7 00 00 00 00 20 00 56 E7 0E E7 E0 E6 B2 E6 84 E6 56 E6 28 E6 FA E5 CC E5 9E E5 70 E5 42 E5 14 E5 E6 E4 B8 E4 8A E4 00 00 00 00 00 00**
* **or to the following: Words @ $E756 [59222]: 0170 E760 E732 0000 0000 0020 E756 E70E E6E0 E6B2 E684 E656 E628 E5FA E5CC E59E E570 E542 E514 E4E6 E4B8 E48A 0000 0000 0000**
* **which looks like an MSCW: MSCW @ $0170 [ 368]: StatLink=0170, DynLink=0170, MSIPC=0DAA, MSENV=E756, MSPROC=1, LocalData[0]=00E6**
* **var1283 is possibly pointing to an MSCW (indirect) in USERPROG.LOAD at IPC = 373, DbgCnt = 2554**
* **VAR1283 gets set in USERPROG.LOAD @ 182 with the value $E756 which might point to an MSCW. MSENV and MSPROC are not set? This “MSCW” address does not match any address on the call stack.**
* **The final exception occurs when trying to store into the VOL\_INFO field on the third pass through the code in USERPROG.LOAD at line 160. The address that it is trying to store to is 5461.**

**2/20/2023:**

* **The STO instruction at 182 does not seem to be using the top 2 items on the stack. Because the SINDO replaces the TOS with something else.**

**2/21/2023:**

* **The EREC at $E70E has gotten corrupted**
* **it is already corrupted before the p-Code even starts!**
* **It is getting mashed in *BuildEnvironmentIV4\_12* in the *Initialize a SIB* code**
* **The menu items in debugger / breakpoints “*changed memory”,* and “*expression true global”* do not work.**
* **Getting Overflow @ 13:USERPROG.INITSCREEN**
  + **Going into bunch of SLDC0 which overflows the stack USERPROG.INITSCREEN is calling READSEG @ 155**
  + **CSP proc number is 39 which comes after the CSPEND (I need to compare this to VPC)**
  + **Its trying to load GOTOXY, SegLeng=74\*2, from Block=89 on unit #4**
  + **After the call to READSEG**
    - **Globals.Lowmem.SEGB is nonsense (Globals.Lowmem.SEGB). The SYSRBOOT garbles it. This is happening because SYSIO is using a base address (UBUF) of 16.**
    - **GOTOXY does not show up in my SEGMAP (only showing the first page of the segment dictionary). I may be able to use LIBRARY.CODE.**
      * **16 FILEOPS 2428**
      * **17 SCREENOP 582**
      * **18 SEGSCINI 508**
      * **19 SEGSCPRO 229**
      * **20 SEGSCCHE 126**
      * **21 GOTOXY 74**
      * **22 REALOPS 2189**
      * **23 SPOOLOPS 184**
      * **24 SPOOLTAS 594**
      * **25 DEBUGGER 490**
      * **26 EXTRALEX 5983**

**2/22/2023:**

* **Highest legal CSPTABLE entry for version 4.12 is 39**
* **The two versions are the same until somewhere before DbgCnt 17,705 → 17,708. USERPROG->OSINIT->ALLOCATE->INCRBASE. Put a break on INITKERNEL. Things match to the start of INITKERNEL. Except that the IPC on XPS is always 1 higher?**
* **The mismatch occurs at a TP in ALLOCATE which follows different paths after the TJP. IPC = ~212.**
* **SYSCOM.FreeSpaceInfo mismatch: $00010000 </> $FFFE0000,**
* **Size: 1254 </> 127**
* **PoolOutside in SYSTEM.MISCINFO isn’t carrying through**
* **Old version uses a checkmark (EnableExternalPool). New version is trying (unsuccessfully) to access info in SYSTEM.MISCINFO which has not been loaded yet.**
* **XPS makes it to DbgCnt 18670 range check error**
* **Error occurs (possibly) in p#52, POOL\_ADJUST called in KERNEL.FAULTHAN IPC=129. Put a break in KERNEL.FAULTHAN IPC=126**
* **Occurring in CheckForBreak GetRelIPC**
* **When executing FAULTHAN, the IPC is wrong as is the procedure name, etc**
* **Old: fSI = 2716, fProcBase = 2590**
* **New: fSI = 68252, fProcBase = 2590**
* **Conclusion: I had to change all of the recent changes concerning word → longword back to the way that they were before.**
* **I also changed CSPEND to 39 to correspond to the VPC version.**

**2/23/2023:**

* **CPL 6 (at 65 in OSINIT) adds a bazillion to the DbgCnt**
* **ReadSegRefs (was FILEPRES) loads the directory at $06C4**
* **SYSTEM,MISCINFO gets loaded at $E3CE**
* **Its stuck in ReadSegRefs (was FILEPRES)**
* **Highest address in ReadSegRefs (was FILEPRES) is 1521**
* **Occurring in the area of 128 to 131 in OSINIT**
* **First it looks for SYSTEM.MISCINFO**
* **Then it looks for SYSTEM.PASCAL and finds it at IPC=223**
* **USERPROG.ReadSegRefs (was FILEPRES) at 205 is comparing IDX to NUMFILES (47..?). Its looking for SYSTEM.PASCAL**
* **I got as far as IPC = 363 in ReadSegRefs (was FILEPRES). It occurs a number of times until DbgCnt = 5300 possibly ignores that fact that it had, indeed, found HEAPOPS?**
* **The EQBYT at 1381 seems to be looking for**

**2/24/2023:**

* **Even though ReadSegRefs (was FILEPRES) seems to have found KERNEL, it keeps on doing comparisons with ? GETCMD, INITOPS,**

**2/25/2023:**

* **Files I may need:**
  + **F:\NDAS-I\d7\Projects\pSystem\Temp\Initial EVEC, EREC, SIB.rtf**
* **Loading a packed field (located at $D7B6) from the segment dictionary (located at $D6F6) -- possibly the seg\_type**
* **SourceCode\_Step is always getting 0 from LineNrFromIPC(RelIPC,fSourceCodeOffsets)**
* **I lost the tracking of a MOV command which was moving a SEGMENT name. Was in ReadSegRefs (was FILEPRES) at IPC=953**
* **I’m getting a range check error when I try to maximize the p-System window which has been minimized. CRTInfo.MaxRows = 50 BUT Length(fScreenBuf) = 25. Dump the CRT info. Need to verify that CRTInfo.MaxRows is plausible and possibly adjust Length(fScreenBuf) to match..**
* **Put a breakpoint into ReadSegRefs (was FILEPRES) @ 953 (already exists) and wait for the DbgCnt to reach 95,883. Step until I get out of the loop.**
* **Maybe consider writing a routine for *Eternal Decoder Window*** **which will dump the EVEC table.**
* **If I do a pure wi-fi connection, I get 100Mbs upload. Maybe it is the cable?**
* **Windows that I have open**
  + **SEGMAP-SYS4Z-SYSTEM.PASCAL.CSV**
  + **How to create DOSBOOT (2023)**
  + **IPCCOUNTS.TXTInitial EVEC, EREC, SIB.RTF**

**2/27/2023:**

* **The call “CXG Kernel 1 1” occurs at DbgCnt 103783**
* **Occurs on the first call to *Restore -* coming from Globals.LowMem.CURTASK which is setting SP to 0.**
* **CurTaskAddr is $154 (340) and the “saved” task that is being restored is at $154.**
* **CURTASK is initially set to $154 (340). It never gets changed (except perhaps when running some other task?).**
* **A TIB is 24 bytes long.**
* **The initial SP (in the TIB) is $E448. The TIB is located at $0154. TIB.SP is located at $15C (348).**
* **The change is occurring in OSINIT at IPC=9 following a call to LPR.**
* **How can the SP in the current task get set to 0 without tripping one of my breakpoints???**
* **Getting an AV error (“attempt to call virtual method on freed item”) when I try to ADD a watchpoint.**

**2/28/2023:**

* **SCIP is at 0c9F. Put a breakpoint there**
  + **cx is the lex difference**
  + **bx is the static link**
  + **MSCW1**
    - **EC0C - static link**
    - **EC0C- dynamic link**
    - **0001 - IPC**
    - **FFD0 - MSENV**
    - **0001 - link count**
    - **0721 - procedure number ?**
  + **MSCW2**
    - **EC0C - static link**
    - **EC0C - dynamic link**
    - **0001 - IPC**
    - **FFD0 - MSENV**
    - **0001- link count**
    - **0038 - procedure number ?**

**12/27/2022:**

**Important locations**

| **Addr (Hex)** | **Contents** |
| --- | --- |
| **0008** | **CS** |
| **000A** | **DS** |
| **000C** | **SI (Saved IPC)** |
| **0012** | **ES (Saved ES)** |
| **0014** | **current MSCW** |
| **0018** | **current ProcNum** |
| **001E** | **TIB?** |
| **0022** | **current TIB** |
| **00B8** | **CurProc (usually?)** |
| **0038** | **IORESULT / SYSCOM** |
| **003A** | **(was APOOLSIZE)**  **(I’m going to use for DbgCnt)** |
| **003C** | **SYSUNIT** |
| **004C** | **GDIRP** |
| **0082** | **CRTINFO** |
| **0098** | **TIB** |
| **00A0** | **tib\_sp** |
| **00A2** | **TIB\_MP** |
| **2212** | **IOR** |
| **00c4** | **EVEC** |
| **00B8** | **CURPROC** |
| **0265** | **GetBigB→ CX (Len)** |
| **CSP Table start at 1688** |  |
| **1348** | **PoolBase** |
| **BX div 2** | **Opcode in FETCH** |
|  |  |

* **See the file \\xps-8930\ndas-i\d7\Projects\pSystem\Docs\2023-NOTES.RTF**
* **I get the changed memory breakpoint at cs:00a0 when the DbgCnt is 0.** 
  + **changes from $0000 to $ddB6 -> $dda6 -> $dd8a**
* **The break is occurring at $3A7C**
* **Delphi: regs.sp changed to 0000 or was that way originally**

**3/1/2023:**

* **Possibly occurring in USERPROG.OSINIT in call to SPR at IPC = 130 (DbgCnt = 102793)**
* **Initial static link set to $170 in LOAD\_PSYSTEM**
* **What is BP pointing to in SPR? ERECp? The static ($E448) points to an all zero area of memory which then sets the SP to zero. Where did the static link get set to $E448?** 
  + **It gets set to $E448 in CPL (from the MP) on the initial CPL**
  + **WordAtp[$E448] is ALWAYS zero.**
  + **It comes from the initial value for MP**
* **Take a look at STEP 11 in Load\_PSystem**
* **Do I have a way to print a list of procedures in a segment? Yes. “Search Volumes”, “Procedure Report” function**
* **The EREC.SIB seems to refer to the KERNEL segment but the ? is still referring to USERPROG. ProcNr is gibberish in (3491) in *ProcNameFromErec***
* **Where was I at when I set the EREC watch points in the Delphi debugger. Possibly NEWSEGMENT OR NEWENV?**
* **The MSCW at $E43A has a garbage value for MSPROC (3491) in DisplayCallStack.**
* **I’m working on the “Dump EVEC, EREC, SIBs” function.**

**3/2/2023:**

* **My ERECs are 46 bytes apart -except- for the first two which are 48 bytes apart**
* **The second EREC should have been located at $E756 - 46 = $E728. I had forgotten to include VECT\_LEN in the addressing.**
* **Segment map dumped a bit later does not match at all.**
* **Does the Map actually refer to segments? There may, in fact, be up to 32 of them which would correspond to the map. Yes.**
* **Do the SIBs need to be in the heap? My memory dump from the VPC would appear to show the ERECs in high (stack) memory.**
* **My expected addresses for the ERECs: E754, E726, E6F8, E6CA…**
* **My expected addresses for the SIBs: E730, E702, E6D4, E6A6…**
* **Where am I supposed to get the SegNr from? TSegInfoArray? No. When I try to use SegNr from the SegInfo field of the dictionary, they all are 1 or 2. Internal architecture guide refers to this segment number as “local” segment numbers**
* **When initial loading has completed, virtually all of the ERECs and SIBs have been changed. Most appear to now be in the heap rather than in the stack.**
* **Getting a “Range Check Error '' in GetRelIPC at DbgCnt 102819. Segments such as DEBUGGER and SCREENOP now appear in the vector.**
* **After disabling the changed code to do BuildEnvironmentIV4\_12, I am getting the apparent UNITREAD(0) which I previously fixed. This may have been related to VOLINFO being empty.**
* **Back to getting a “Range Check Error '' in GetRelIPC at DbgCnt 102819.**
* **Dying within CalcProcBase because it is getting a SEG of 0 (and a ProcNr of 1). DbgCnt = 102643 (in the debugger in DisplayCallStack).**
* **Seg\_Pool = 0, SegBase = 0 → SEG = 0**

**3/4/2023:**

* **Currently getting a “Range Check” in CalcProcBase when assigning something. DbgCnt = 102643. ProcPtrOffset may be wacky.**
* **Back to UnitRead(0). Be careful to boot version 4.12!**
* **SysCom says that POOLOUTSIDE is false (booting from F:\NDAS-I\d7\Projects\pSystem\Volumes\system4z.vol). Bad PoolBaseAddress (6, 11360). When looking at the VPC version of SYSTEM4Z, I also see POOLINFO.BASE[0] = 1, POOLINFO.BASE[1] = $54C0 which is the same as what SETUP.CODE sees when running on PSYSTEM.EXE o SYSTEM4Z.VOL. I copied over SYSTEM4Z.VOL from VPC to Delphi and replaced the previous version which was displaying messed up *Pool Base Address*  info.**
* **I’m still getting *integer overflow*  in *CalcProcBase.* The debugger is trying to display the dynamic call stack. Called from *IPCWithinProc* . Both SegPool and SegBase in the SIB are zero. The environment vector seems plausible.**
* **The SYSCOM page in my debugger is still showing PoolOutside = false, PoolSize = 0, PoolBase = 0, and Resolution = 0**
* **No breaking at OSINIT, IPC=121 for some reason?**

**3/6/2023:**

* **None of my breakpoints in OSINIT are breaking - presumably because RELIPC is bad.**
* **Things go crazy when exiting ReadSegRefs (was FILEPRES)**
  + **ReadSegRefs (was FILEPRES) is doing a RPU 1699 (words to cut off of stack)**
  + **RelIPC jumps to 3419**
  + **ProcBase is 6?**
  + **RCE after IPC = 953**
* **When entering ReadSegRefs (was FILEPRES):**
  + **'SegBase @ $E7A0 [59296]: , DictOffs= 0E42, RelocOffs= 0000, SegName= USERPROG, Sex= 0001, CPOffset= 0DAE, NrReal= 0000, RealSize= 0004, RealSubpOffs= 0002, NrProc= 000E, ConstPoolSize= 120'**
* **DumpCodeSeg**
  + **ProcPtrOffset: 32**
  + **ConstPoolOffset: 3502**
  + **NumberOfProcedures: 8737**
  + **AddrOfNumberProcedures got changed from $F5E2 to $E7C0**
  + **RelIPC gets dinged between DbgCnt=160 and 173?**
  + **DbgCnt = 160 occurs in “FLIPSEG”**
  + **DbgCnt = 173 occurs after exiting back to OSINIT**
  + **CalcProcBase frequently gets passed an address of 0.**
  + **This occurs in a call to IPCWithinProc**
    - **IPCWithinProc is called with an ERECAddr of $E754**
    - **The EVEC is at $E75E which looks plausible**
    - **Its links to an EVEC, SIB both look plausible**
    - **But the link to the SEGBASE (0000) is not.**
    - **UserBase is $E7A0**
    - **RelIpc = 835 is not reached *prior* to NumberOfProcedures or ProcPtrOffset going wacky. Changed the breakpoint to 831.**
    - **GLOBALS.LOWMEM.SEGb never changes**
    - **ProcPtrOffset got changed from $0E42 to $0020 as ProcNumber 9→6, RelIPC > 835, DbgCnt > 6900+ and < 6942, Crashes after DbgCnt - 6940 (which may be occurring after a MOVELEFT).**
    - **The MOVELEFT is moving 2 bytes from $E19C to $E7A0. $E7A0 is the address of the segment info. This move is going to mash the “proc dictionary pointer” (first word of the segment dictionary).**
    - **My “Backup DB to text files” is backing up to bad locations (i.e.: “IV.22: 376 .pas, 306 .pcode, 341 .VarList files were copied to”) because “Text Backup Root Path” was empty. Currently they are all getting put into “Volumes”**

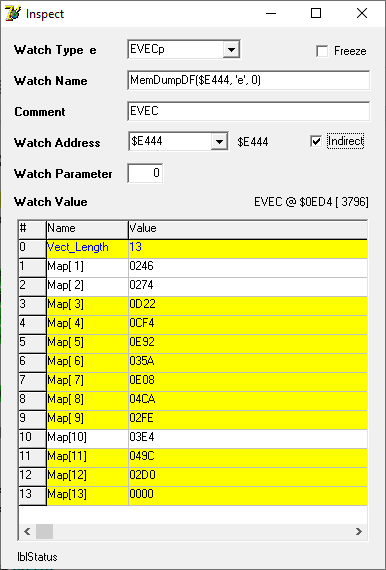
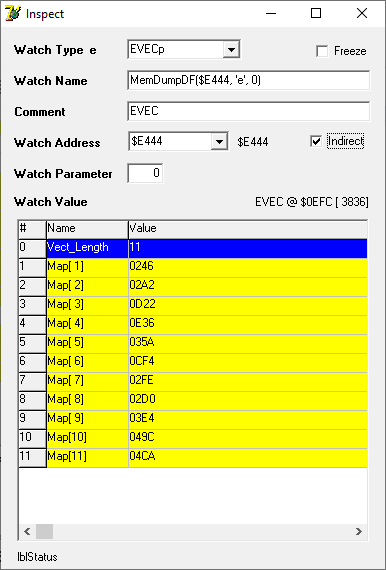
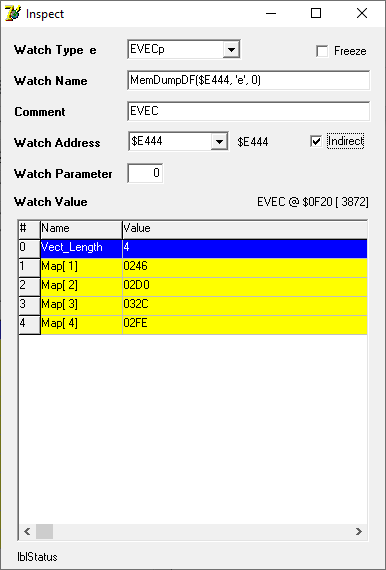
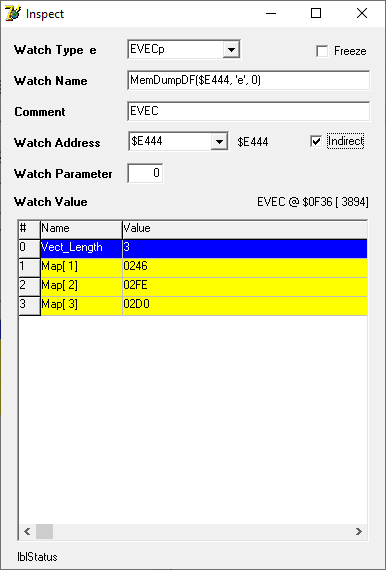
**3/7/2023:**

* **Should be ERECS & SIB actually be located in the heap?**
* **DatabaseList.Count = 2?**
* **How do I find the top of the heap?**
  + **see: HeapInfoFormat (wt\_HeapInfo)**
  + **“allocate space on heap for segment”**
  + **right now I am assuming that I can locate it in low memory and protect it with SP\_LOW**
* **All of the EVEC, EREC & SIB info which I constructed initially, gets wiped out**
* **GetWordAt passed an ODD address @ #6:USERPROG.ReadSegRefs (was FILEPRES), Ofs: 530, DbgCnt = 41588**
* **lcl1383 has something to do with the starting MSCW**

**3/8/2023:**

* **Directory loaded to $0CC6 at DbgCnt 236 in USERPROG.ReadSegRefs (was FILEPRES).**
* **This is in ReadSegRefs (was FILEPRES).**
* **$0CC6 is a VOLINFO record. But the directory is loaded to $0CC6?**
* **SYSTEM.MISCINFO loaded at $E71C (on the stack) DbgCnt 346, USERPROG.ReadSegRefs (was FILEPRES) IPC = 135**
* **$0170 is address of Env data**
* **Getting the AV following the freeing of the DebugWindow (SetReference)**
  + **Occurs when pCodeDebugger is doing a SetReference(false) during a DestroyComponents. Instead of calling FreeAndNil(frmDebugWindow) during the notification. I just set frmDebugWindow to nil.**

**3/9/2023:**

* **The DashBoard shows the EVEC at $0204 but my map lists it at $01FA which is a difference of 10. bytes? Miscalc?**
* **Back to SETWORD passed an ODD address at DbgCnt = 41588, 6:USERPROG.ReadSegRefs (was FILEPRES), IPC=529**
* **EVECs built: DBGCNT = 25812, 33964, 33761, 38793**
  + **$0ED4 completed at DBGCNT = 25812 (28 bytes)**
  + ****
  + **$0EFC completed at DBGCNT = 33964 (24 bytes)**
  + ****
  + **$0F20 completed at DBGCNT = 33761**
  + ****
  + **$0F36 completed at DBGCNT = 38793**
  + ****
  + **Odd address error at DBGCNT = 38910**
* **Try to put a breakpoint whenever the EVEC address at $E444 changes. Then put a breakpoint just BEFORE it gets changed and look at the current EVEC. See if I can figure out what segment is being worked on. Does the address for the EVEC look like it is properly located?**
  + **Gets changed at** 
    - **ReadSegRefs (was FILEPRES) IPC=745. Goes from nil to $0204 which shows an empty EVEC (which is the original EVEC from load time)**
* **All of the EVECs created have $0246 as the EREC for Map[1]**
* **Other ERECs may appear in more than one EVEC**
* **Check the breakpoint at ReadSegRefs (was FILEPRES) IPC=650, LDO 206. Where does this get set?**
* **Also getting different DbgCnt for the ODD address exception. Latest is 38910.**

**3/10/2023**

* **Some of the ERECs in the EVECs seem to be valid and link back to valid EVECs and some do not.**
* **Temporarily reverting to the 20230306 version. Getting the RCE in GetRelIPC again. DbgCnt = 102853**
* **The “Pool Base Address” seems to be wacky (87232 = $000154C0)**
* **NEWWAY: Got back to functional by switching back to F:\NDAS-I\d7\Projects\pSystem\Volumes\SYSTEM4.VOL (still getting the odd address error in ReadSegRefs (was FILEPRES) at IPC = 669**
* **OLDWAY: back to RCE in GetRelIPC. Even though DbgCnt is incrementing, it is not displayed in the debugger (because the debugger is crashing with the RCE).**

**3/13/2023**

* **ERangeError DbgCnt = 103205**
* **'103204=SCXG1 (USERPROG.INITSCRE @156),103203=SLDL1 (USERPROG.INITSCRE …**
* **I am getting invalid number of procedures 8337**
* **Trying to READSEG to address $154D0**
* **F:\NDAS-I\d7\Projects\pSystem\Volumes\system4z.vol (OldWay)**
  + **When breaking at DbgCnt = 102,842, the EVEC at $0ED4 looks pretty good**
  + **The EVEC at $0F36 looks pretty good**
  + **The EVEC at $0F20 looks pretty good**
  + **The EVEC at $0EFC looks pretty good**
  + **The EVEC at $1208 looks pretty good**
  + **$0F7A looks pretty good**
  + **$0FD6 ok**
  + **$102A ok**
  + **No EVEC at $0204**
* **At a glance, all of the ERECs look plausible. See the file: C:\temp\ERECList.txt**

**3/14/2023:**

* **USERPROG has a 4 byte mis-match in *Code\_Len* 1826 <> 1832?**
* **ALL of the other *CODE\_ADDR* and *CLODE\_LENG* match exactly.**
* **The only *SegBase* that is non-zero is for GOTOXY AND IT IS $0010.**
* **The final UNITREAD is** 
  + **ODS: CallIO: UNUM=4, UBLK=89, ULEN=148, UBUF=154D0**

**This appears to be trying to load GOTOXY.**

* **The *DataSize* on KERNEL does not match: 0 <> 253**
* ***NextSort* and *NewLoc* are always the same because they are stored in the same location.**
* ***NextSIB* and *PrevSib* are always zero.**
* **Residency is always zero.**
* **The Kernel SIB shows DataSize as zero. Should it actually be 253? I have changed BuildEnvironmentIV4\_12 to initialize the SIB from data dictionary (253).**
* **SEGMAP shows Data\_Size for USERPROG as 0. DashBoard EVEC chain dump shows it as 17739**
* **GOTOXY is the only segment which has a non-zero value for SEGBASE (0010). Implication: only GOTOXY is loaded.**
* **BOOTERROR only got called 5 times**
* **Remember to find an alternative for *ProcBase.***
* ***Watch ProcPtrOffset***
* **SEGTOP ($0E42) does not seem to agree with the base ($E7A0)?**

**3/15/2023:**

* **Address of “number of procs” $E7C0 - ORIGINAL**
* **Globals.LowMem.SegB at $2A. DbgCnt 102678**
* **The number displayed for SEGTOP in the debugger is wrong.**
* **DbgCnt = 102568 when entering GETPSEUDOSIB**
* **When does ProcPtrOffset get changed?**
* **ProcPtrOffset is stored as a word offset. It must be changed to a word offset.**
* **USERPROG.ReadSegRefs (was FILEPRES) LINE 1318 is overwriting the “Number of Procs” field of the Segment Base Info.**
* **Delphi started to get wacky- refusing to process “Step”, “F9”, etc. I rebooted the PC and rebuilt the project to try to fix it. It did not work.**
* **Tried removing run-time (debugger) MemDumpDW. No go.**
* **Tried commenting out ALL of the function calls in RepaintDisplay. The Delphi debugger watches are already not being updated immediately after the dashboard is loaded.**
* **Removed the DebugWindow completely. The problem still exists even by the time that the first call to ProcCall is made.**
* **Even PhotoDB is having the same problem.**
* **Finally traced the problem to the Windows “Virus and Threat Protection Settings” having “Real Time Protection” turned on!**

**3/16/2023:**

* **Delphi debugger working fine today. A Windows update occurred last evening.**
* **What is 16 words or 32 bytes in size? Possibly:**
  + **TSegMiscArray**
  + **TSegTextArray**
  + **TSegInfoArray**
* **How did the segment get loaded to $E7A0? CallIO does not show it! I am loading it in DebugInterpreter. I must not be loading it to a safe place. It is being loaded to USERBASE (on the stack). UserBase is not saved.**
* **Look at *Seg\_Base* in the SIB.**
* **I changed the initialization code to set SEG\_BASE (for the Kernel only) in BuildEnvironmentIV4\_12 (OldWay) to the address where the segment gets loaded ($E7A0).**
* **The revised code now smashes the ProcPtrOffset rather than the AddrOfNumberOfRealConstants. I think that I must be loading the Kernel to the wrong location. Currently getting loaded to the stack. Maybe it should be loaded onto the heap.**
* **The initial EVEC is getting loaded just BELOW USERPROG/KERNEL.**
* **Maybe it is the EVEC, ERECs and SIBs which should be loaded onto the heap?**
* **USERPROG is getting loaded into high memory (on the stack)**
* **Does seg\_leng get stored in the EREC/SIB for the Kernel?**

**3/17/2023:**

* **After changing the EVEC, EREC, SIB initialization code to the NEWWAY2 version, I can make it as far as DBGCNT = 38910 before getting an odd address error occurring at IPC 669 in USERPROG.ReadSegRefs (was FILEPRES). At least the procedure dictionary is not mangled 🙂**
* **But now it is smashing the "Env\_Data" in the Kernel EREC. This occurs at DbgCnt = 6941. It may be attempting to set the VECT\_Length of some EVEC.**
* **Where does 1389 get set?**
  + **Put a break into ReadSegRefs (was FILEPRES) at 1281 and 1383. Be sure to have watch window set for the local variables in “ReadSegRefs (was FILEPRES)”**
  + **Local 1389 is already set before it is referenced in ReadSegRefs (was FILEPRES).**
  + **Actual address is $E51A.**
  + **Gets changed in ReadSegRefs (was FILEPRES) AT LINE 1197.**
  + **IXA claims that the index is 129 (offset into last block of segment).**
  + **Why does var1375 have a 129 (=$81) in it? Where does it get set? Actual address is $E4FE. This is the value for *WordsInLastBlock* .**
  + **Do I need to be initializing the SEG\_LENG field of the SIBs? I think that I am doing that. How can I verify? Pretty sure that they are being initialized.**
  + **At line 1134 in “ReadSegRefs (was FILEPRES)” it is reading something from unit #4 starting at block 14 which ought to be the first block of a segment to $E51c but it does not look like a segment dictionary. Loading the last block of the segment (I think).**

**3/18/2023:**

* **Is actually reading block 14 (as was requested) but block 14 seems to be the wrong block. Not if it is trying to load the last block of the segment.**
* **The block number comes from address $E500.**
* **The block number gets set in ReadSegRefs (was FILEPRES) at line 1080.**
* **I should be able to see the TSeg\_Family\_Rec using MemDump(Addr, wt\_SegDict)**
  + **'data\_size= 253, seg\_ref\_words=75, max\_seg\_size=16, text\_size=29'**
  + **seg\_ref\_words div 5: Why? Because each Seg\_Ref\_Rec uses 10 bytes or 5 words**
* **See the notes continued in the p-Code Source USERPROG\_ReadSegRefs (was FILEPRES).pcode**
* **ReadSegRefs (was FILEPRES) line 1278 - Is this planning to write to the word AFTER the EVEC?**
* **Prior to the MOVELEFT:**
  + **'Words @ $DA2E [55854]: 0002 0000 0246 0000 E4EC**
* **Moving 2 bytes from $E4EC to $0246 (which smashes the EREC.Env\_Data at $0246 in the original EREC). What is located at $0E4C? $0E4C is the “MaxSegs” field in the local variables.**

**3/22/2023:**

* **Version 4.2 cannot mount subsidiary volumes**
* **Version 4.2 is not repainting the screen properly when in the editor- lines disappear until you do a Verify.**
* **The editor adjust command is not working right.**
* **STRINGFILE = FILE OF STRING[30];**
* **The goal right now is to be able to execute the MAINTAIN.CODE** 
  + **cannot locate UNIT READS.**
    - **see MLIB.text on PSYSTEMX.VOL**
  + **cannot locate UNIT SCREEN**
  + **the volumes that may be needed include:**
    - **RL0, INS:, OLDGF:, YEARLY:, JUNK:**
* **and then UTILITY.CODE program**
  + **L=UTILCOD:ULIB.TEXT**
  + **Still getting I/O error: file not found, Seg INITIALI p#7 O#25**
    - **going to try mounting the missing volumes. That did not work.**
    - **Copy INS.SVOL, YEARLY.SVOL,**
* **and Medoffice**
* **Next make the DumpSys program work**

**3/23/2023:**

* **For some reason the editor is trying to write <esc>I which causes problems.**
* **This is the “Insert Line” sequence for a VT-52. This sequence does not appear in SYSTEM.MISCINFO. I hard-wired the assumption that the <ESC>I means “insert line”. This Advanced System Editor has its own configuration program (ASS.CONFIG.CODE) which simply ignores the usual way that SYSTEM.MISCINFO stores this stuff and uses its own settings stored elsewhere in SYSTEM.MISCINFO. If I knew what format this information was stored in, I could use the size of SYSTEM.MISCINFO to get the proper settings.**
* **A “Range Check Error” kills the program w/o even a message.**

**3/24/2023:**

**Sequence:**

1. **Enter FilerMain**
2. **Start the debugger**
3. **Close the debugger**
4. **Do ^U to view contents of a volume**
5. **This generates an AV when the volume tries to call PutIOResult to an interpreter that no longer exists.**

* **I’ve put procedures that should be global relevant into DUMPSYS:MYUTILS.TEXT**
* **Other globals specific to the TESTPROC.TEXT program have been put into UGLOBALS.TEXT**
* **The library file MYLIB.TEXT tells where to find the units.**
* **To Test:**
  + **Make sure that the JUNK: volume is mounted**
  + **Boot the V4.2 system**
  + **XP=DUMPSYS:**
  + **XL=MYLIB.TEXT**
  + **XI=B // to recompile everything**
  + **XTESTPROC**
* **The SYSTEM DATE is always wrong**

**3/27/2023:**

* **The tab key is echoing when it should not**
* **Each new line of the word dump is advancing too far**

**3/28/2023:**

* **Memory dump doesn't match expected. Fixed.**
* **Getting an ODD address error when I try to toggle**
* **When it breaks entering DUMPSYS.DUMPSYS (supposedly), the ABSIPC IS $1EE6A.**
* **I am getting ERANGEERRORS -- perhaps related to LONGWORD stuff**
* **Also error in unresolved in SegDictFormat**
* **Why does it think that AlphaFormat is not defined?**

**3/29/2023:**

* **GetWordAt passed an ODD address $007D at 2: KERNEL.EXECERR, Ofs: 2**
* **Memo1 (pcode) starts off assuming that it should be dumping from location 0.**
* **The call stack is flakey:**

**Dynamic Call Stack Report. Generated on 03/29/2023 2:46:38 PM**

**MSCW Proc @IPC**

**FCD6 2: KERNEL.EXECERR 0**

**FD1E 9: DUMPSYS.DUMPMEM 454**

**FF60 8: DUMPSYS.DISPMEMO 73**

**FFC6 1: DUMPSYS.DUMPSYS 54**

**FFD0 1: DUMPSYS.DUMPSYS 10**

**FFDA 51: DUMPSYS. 40**

**FFEA 1: DUMPSYS.DUMPSYS 5**

**FFF4 2: DUMPSYS.GETFUNC 6832**

**170 1: DUMPSYS.DUMPSYS 3038**

* **Addr := (p+Offset+I) div 2; is messing up Addr**

**3/30/2023:**

* **DUMPSYS does not have any dates**
* **In fact, none of my volumes show dates.**

**3/31/2023:**

* **Searching for the Hex string “48 4a 4b 43”** 
  + **finds it at location $0125 from delphi**
  + **shows it at location $00A5 in DUMPSYS**
  + **shows it at $0125 in in DashBoard**
  + **The difference is 128 ($80) ???**
* **Algorithm to merge source code into p-Code:**

1. **(this assumes that the p-Code perfectly matches the source code- the p-Code should probably be regenerated in the “External Decoder Window”**
2. **Scan the p-Code stream and delete ALL of the comments (no longer necessary)**
3. **Read the p-Code and the source code line by line**
4. **Whenever the line number matches between the p-Code and the source code, copy the Source code line into the p-Code stream as a comment (along with any following lines having the same IPC) and THEN copy the p-Code line.**
5. **if the pCode IPC is less than the latest SrcCode IPC, just copy the pCode line to the output.**
6. **Display the merged p-Code stream in the decoder window and let the user copy and paste it back into the p-Code window.**

* **Working on *MergePCodeWithSourceCode* in *DecodeWindow***

**4/3/2023:**

* **Why do I have two separate locations for *loc* ?**
* **When paging backwards from $0030 by $0080 the result is $FFB0 which is “larger” than I have defined for a word 32767 ($7FFF).**
* **ALL RTE errors are showing up as “Unknown run-time error”**
* ***Merge* is buggy - see (for example) DUMPMEMOry. Source code lines not being placed above the p-Code lines. Probably because the source code listing is obsolete.**
* **When I manually enter the address $0030, even though the addresses display properly, the memory displayed is entirely 0.**

**4/4/2023:**

* **Getting “Error writing file, not enough room” when compiling DUMPSYS. This appears to be occurring in BACKEND5.TEXT.**
* **The final write may have been to :**
* **ODS: CallIO: UNUM=5, UBLK=2, ULEN=754, UBUF=$3652, RESULT=0**
* **i.e., to the directory of DUMPSYS: (but the IORESULT = 0) and the length is very strange? It does not occur if no compiler listing is selected. Error 402 (\*SYSTEM.INFO) or 404.**
* **REMEMBER THAT I HAVE CHANGED THE ACTIVE DATABASE TO VERSION IV.12 !**
* **WRITECODE could be procedure 35. NO.**
* **changing the debugger settings from the debugger does not seem to be remembered.**
* **I cannot find WRITECODE (compiler) in the 4.12 DB. Found in BACKEND5.TEXT in volume IV13COM.VOL.**
* **If I try to edit BACKEND5.TEXT in UEDIT, it doesn’t all get written back to the source file. Cannot just do ^E to edit since the TEXT file transfer back to the p-System will make amess of SYSTEM.COMPILER.**
* **The write error IS occurring in BACKEND5.TEXT when WRITECODE calls IOERROR following a failed BLOCKWRITE. What does BLOCKWRITE get translated to in the interpreter call? It doesn't seem to matter where the listing file is getting sent to.**
  + **I can try to recompile the compiler from IV13COM.VOL**

**4/5/2023:**

* **Remember that I cannot mount SVOLs**
* **To get the compiler to compile (located on IV14COM: and using OPSYS:)**
  + **Compiled using Pascal Compiler [4R0.20-4]**
  + **Compiling version IV.13 c6t**
  + **I had to make some small edits to BACKEND4**
  + **I had to compile (to OPSYS:)**
    - **ERRHANDLER**
    - **DEBUG.SEED**
    - **BACKEND**
  + **I used LIBRARY.CODE to merge the output from compiled FRONTEND with (previously compiled) BACKEND.CODE, ERRHANDLER.CODE and DEBUG.SEED.CODE**
  + **The newly compiled compiler seemed to have trouble with “PMACHINE” and other things (such as “Byte”).**
  + **The old compiler has a problem if the PMACHINE() function is used.**
  + **Display registers says that there is an EVEC located at $24F0.**
* **Remember to change back to OLD.COMPILER**

**4/6/2023:**

* **I need to implement Break key**
* **Dumping a SIB makes it as far as WaitAt(11) which precedes the call to ?. Finally gets an unknown runtime error in MYUTILS P#4 o#16. Never makes it to 19.**

**4/7/2023:**

* **The CALL stack shown in the debugger is a mess**
* **BuildDebugDatabase does not properly handle LCO when decoding**
* **working on disptib**

**4/8/2023:**

* **The system date is always wrong**
* **I need to fix the crash that occurs when you try to close the screen while it is waiting for input.**
* **It would be nice if the arrow keys would auto repeat**
* **SC\_init?**

**4/10/2023:**

* **The CONTROL parameter of UNITWRITE has flags which may be applicable, i.e., “Special Character Handling” (EOF), etc.**
* **Note handling of -2: Is this treated as TRUE or FALSE? It appears to be treated as FALSE (because bit 0 is 0). See NOATTEMPT which does a LNOT on 1 giving $FFFE.prompt**
* **break in sc\_getc\_ch at the call to read(KEYBOARD, CH);**
* **What is r\_leng? Its value is 0.**
* **The keyboard.prefix seems to be something screwy**
* **The function WriteBare doesn't seem to be writing what it should be writing! No. It is actually writing but possibly getting written over.**

**4/18/2023:**

* **Linkages:**
  + **DUMPSYS**
    - **Registers**
      * **EREC**
        + **SIB**
        + **EVEC**
        + **SegBase**
        + **Addr**
* **failed: SIB → NEXT SIB**
* **EREC → SIB gets the address wrong**
* **“Known locations”**
  + **TIB @ $02a2**
  + **EREC @ $24c2**
  + **EVEC @ $24F0**
  + **SIB @ 24CC**
* **trying to display a TIB displays garbage**
* **cannot do links from a TIB**

**4/19/2023:**

* **PSYSTEM.EXE won’t let me do a 50 line page**
* **Copy PSYSTEM.CFG from the laptop**
* **How can I get a copy of PSYSTEM.CFG that doesn’t try to mount everything? Use UltraEdit to edit PSYSTEM.CFG.val machine. There are notes about PDYSTEM.CFG above!**
* **Possible ways to restore:**
  + **E:\Maxtor\Virtual Machines\Windows XP**
  + **E:\Users\Daniel Dorrough\VirtualBox VMs\Windows XP**
  + **H:\Maxtor\Virtual Machines\Windows XP\VirtualBox VMs\Windows XP**
  + **H:\VirtualBoxBackup**
  + **----**
  + **H:\Maxtor\Virtual Machines\Windows XP**

**4/20/2023:**

* **Remember Windows “File History”**
* **Why am I seeing an older version of DUMPSYS.VOL? Because I didn’t copy if from Z: to psys**
* **I need the latest version of search.code on the VPC version**

**4/21/2023:**

* **On V4.12:**
  + **TIB for the currently running task is at $0098**
  + **SP\_low= $75E6**
  + **SP\_upper = $FFFE**
  + **SP = $E952**
  + **Many of the ERECs ($FD4C → $FFD0) appear to be in the stack? (Or maybe “above the stack”)?**
  + **Stack overflow occurs if SP was to goe below SP\_LOW**

**4/22/2023:**

* **How do I find SYSCOM? Starts at $E6 in version 4.2.**
* **Can I dump the segment base info?**
* **I need to implement the break key**
* **I’ve got to fix default system dates**
* **Virtual box finds the value 4 at $3C which would put SYSCOM at $3C-4 = $38? This does look like a plausible syscomrec.**
* **After exiting FILER.EXE, seems like ^E (is still disabled) (cannot reproduce)**
* **It might be better to just use the official GLOBALS.TEXT (turns out that this was too hard and I abandoned the idea)**
* **I should be able to allow “cut, copy and paste” to the p-System window??**

**4/24/2023:**

* **I have suddenly started getting errors (Integer Overflow) when I boot. Occurring in “SCAN”. Probably because I was trying to boot SYSTEM4. . PSYSTEMX seems to boot OK.**
* **FILER is always coming up on the SURFACEPRO built in screen? Somehow, FORMSHOW got disconnected.**
* **Latest saved version of “InterpIV.pas”: F:\NDAS-I\Source Code\Other Sources\pr20230410\pSystem\Src**
* **I get a “Range Check Error” when I try to compile GLOBALS.TEXT.**
* **When I tried to change *SrcO* and *StartAddress* to *LONGWORD* in SCAN, I get conversion errors on “Len := SrcO - StartAddress”. I put them back to just type *WORD*.**
* **I should change GLOBALS.LOWMEM.SYSTEM to SysComPtr^ everywhere because version 4.12 has SYSCOM located at a different address (see above) $38 which is 16 bytes lower.**
* **The boot list would be nicer if recently used stuff appeared near the top**
* **Going straight to SYSTEM HALT!**

**4/25/2023:**

* **Hex number being read incorrectly ($231a being read as $231E) (somehow getting 4 added to it?) The number getting passed to bothways (from prefixinfo?) is incorrect?**
* **The number getting passed into PrefixInfo (8990) is incorrect.**
* **These might have been related to not recompiling MYUTILS?**

**4/26/2023:**

* **PrefixInfo not displaying in the correct location**
* **In the debugger, display disabled breakpoints in yellow**
* **Control makes it to IPC=60 after displaying the PrefixInfo**
* **AddDec for the VECT\_LENGTH writes on top of the PREFIXINFO**
* **wt\_EVEC displays as if EVEC\_LENGTH (12) is the address rather than the contents. Problem was that viewport was not passed as a VAR to WriteBare**
* **InitViewPort should be called BEFORE the FORMAT routine is called-- not internally to it.**
* **EREC is not displaying the SIB**
* **Working on FormatSyscom/Viewports (IncRowNr)**
* **Would be nice to use pSystem debugger info for my debugger**

**4/27/2023:**

* **processor\_types[]**
* **pMachineVersions[]**
* **In version I.5, THEDATE is located at 67.**
* **In version 4, DLASTBOOT seems to be used**
* **Trying to use DLASTBOOT on the volume directory entry?**
* **Changes to PSYSVOLUMES have not been tested.**
* **Initial checkmark in Directory to grid is wrong.**
* **Directory Grid- sort by date does not work**
* **Sometimes the grid date column appears as this (01-Jan-2023) and sometimes as this (20230101)? When initially a “date” sort is requested.**
* **auto-repeat on cursor keys**

**4/28/2023**

* **$E6 (syscom) address found at $017A on Delphi (Version 4.2)**
* **$38 (syscom) address found at $0EF8 on VPC (Version 4.12)**
* **I need to allocate space for syscom in low memory.**
* SEGFAULT sets:
  + message\_sem = (1,0)
  + real\_sem = (0, 674)
  + fault\_message = (340, 3474, 0, 128)
* DI := Integer(@TGlobals(nil^).Lowmem.SyscomSpace.fault\_sem.real\_sem);
  + DI = 244 ($E6+14=244)
* **SyscomPtr must deal with both Version 4 Syscom and pre Version 4 Syscoms.**
* **Compare versions:**
  + **Parameter1 = full file name of program to compare**
* **I am getting an execution error in USERPROG.INITCUP0 - (stopped happening for no reason?)**
* **CallIO getting called on UNITS 5, 6, … with a UBLK = 0 and ULEN = 0**
* **XI=BL does not work? (but XI=BL.TEXT does work?) (no longer true?)**
* **Problem occurring when trying to restart the p-system without exiting everything.**

**5/1/2023:**

* **Going to try to make DUMPSYS guess which version of the operating system it is working by examining the UNITNUMBER field of SYSCOM and looking for units 4, 5, 9, 10. If a syscom (located at either $E6=Version 4.2 or $38=version 4.12) containing a valid unit number if found, that will be used for the version number.**
* **When the PRINTER: file gets closed, it would be nice to list it on the FILER screen**
* **Unit2(systerm:): Pressing <tab> in the editor sends a string of 7 blanks followed by 1 blank,**
* **Unit1(console:): Pressing <tab> echoes <tab><tab>**
* **TfrmPSysWindow.FormKeyUp is where the TAB gets written?**

**5/3/2023:**

* **raising an exception in PutIntoBuffer never makes it back to FETCH-- i.e., trying to exit back to FETCH when the user presses the break key.**

**5/4/2023:**

* **MOUNT calls FMOUNT**
* **FMOUNT is contained in FUTIL.TEXT (on P4021FA.VOL)**
* **FUTIL.TEXT is an include file to MAIN.TEXT (which is really FILEOPS) which is on volume P4021FA.VOL**
* **Compiling requires that OPSYS.VOL be mounted.**
  + **LUNIT = 69?**
  + **Is the Unittable really located at $1714? Yes. I think so.**
  + **Why is the UnitTable always showing up as completely blank? Because Param was always zero.**
  + **After “mounting” DOSSRC.SVOL, it does appear in the unit table.**
  + **The only entry that was showing up in the UnitTable was PSYSX? Somehow (magically?) the whole table appeared. (maybe following a Volumes command in the FILER?)**
  + **Maybe $1714 isn’t the real UnitTable.**
  + **Maybe Strings stuffed into pSystem records don’t always have a trailing byte.**
  + **Only units thru 13 are being displayed.**
  + **The UnitTable doesn’t have anything in it until I do a Volumes command in the Filer.**
  + Utable[ 30]: UTablEntry @ $19E4 [ 6628]: Uvid= , UEovBlk=32767, uphysvol=11, ublkoff= 482, upvid=UTILITY
  + **The mounted subsidiary volume doesn’t show a volume name (UVID) which may be why it does not appear in the Filer volumes list.**
  + **The Dashboard display doesn’t automatically update.**

**5/8/2023:**

* **19**
* **Moving 6 bytes from $FEA4 to $19E4 (“DOSSRC” → “DOSSRC”)**
* **UnitTableFormat expects the TABLE address-- not the ENTRY address**
* Utable[ 30]: UTablEntry @ $19E4 [ 6628]: Uvid=DOSSRC ,UIsBlkd=true,UIsSpecial=false,SpecialBuf=false,StdaMap=false,UIsLocLocked=true,UEovBlk= 400,uphysvol=30,ublkoff=82,upvid=
* **everything looks OK when it gets to offset 308 in the fmount p-Code**
* **But, after exiting MOUNT the UVID has been cleared. UnitTableEntry is located at $19E4**
* **UnitEntry modifications:**
  + **FILEOPS.FMOUNT sets the UVID at IPC 258**
  + **FILEOPS.FETCHDIR clears the UVID at IPC 32**
* **My debugger shows DOSSRC.SVOL and COMP.SVOL as mounted but p-System does NOT. Look at references to *ublkoff.***
  + **After doing a V(olumes command, the UnitTable gets populated.**
  + **These volumes are fetched 1..30**
  + **FETCHDIR ipc=27 clears uvid**
  + **Its doing a UNITREAD on UNIT 30 but using block 2 as the directory block and consequently is not reading the SVOL directory.**
  + **Look at the assembler version PME-DEBUG**

**5/9/2023:**

* **screentest.code not found in search of all mounted volumes**
* **“Not enough room” to edit SEARCH.TEXT**
* **Why is it looking at unit #128? FILEOPS.FOPEN IPC=24. UNITCLEAR on device 128 is used to indicate that the “bridge” is open.**
* **Why is MemDumpDF listed as “unknown”? Because I’m using the non-debugger version of Filer.**
* **My SEARCH.TEXT is not scanning the subsidiary volumes. Maybe because UNITREAD is not processing subsidiary volumes. Was not true.**
* **SEARCH.TEXT is located on F:\NDAS-I\d7\Projects\pSystem\Volumes\15SYS1.VOL**

**5/11/2023:**

* **Change SyscomWhat to change when tab is changed**

**5/15/2023:**

* **<PgDn> key is echoing**
* **FormKeyDown**
  + **Printable Characters fire it**
  + **<PgDn> does not**
  + **<PgUp> does not**
  + **<End> does**
  + **<Insert> does**
  + **<Delete> does not**
  + **<Ctrl><right> does**
  + **<Ctrl><left> does**
  + **<home> does**
  + **<esc> does (with an echo)**
  + **<next> sets the key to 0 in FormKeyUp**
* **None of the cursor keys are working in the editor now.**
  + **FormKeyUp seems to get fired on the arrow keys**
* **I got the arrow keys to work again by re-enabling vk\_up, vk\_down, vk\_left and vk\_right in FormKeyUp**
* **Could this have something to do with MenuItem.ShortCut()? Yes. It does.**
* **Need to fix BACKSRCCODE.BAT**

**5/16/2023:**

* **vk\_Next detected (and handled) in FormKeyDown**
* **getting duplicates on <PgUp> and <PgDn>**
* **Try putting Key := 0 into vk\_Next case in FormKeyUp**
* **Arrow keys and paging keys up at 10:12 on 5/16/2023**
* **SetFunctionPrefixes is called in MiscInfoUnit**
  + **SetFunctionPrefixes is defined in CRTUnit.**
  + **It depends on the VersionNr**
  + **Maybe SetFunctionPrefixes should be defined in CrtWindow and Overridden in PSysCindow**
  + **Or maybe it should be defined in the MiscInfo unit**
* **DefaultKeyInfo wants a VersionNr for when it creates CrtInfo**
* **LoadCrtKeyInfo**

**5/17/2023:**

* **How did “$Delphi” environment get set to “SurfacePro”**
* **FunctionPrefixes not set**
* **DefaultCRTInfo creates TCRTInfo**
* **Creation order:**
  + **TfrmPSysWindow.Create vn\_VersionIV**
  + **TfrmCrtWindow.Create {no version nr}**
  + **fCrtInfo created {vn\_Unknown}**
  + **When does TfrmPSysWindow set the version number? It sets it AFTER the inherited stuff where it was used prematurely.**
  + **DefaultCRTInfo call should be delayed until after the window has been created**
  + **If CRTInfo isn’t created before MaxCols, MaxRows (etc) it will cause an AV**
  + **I make it as far as the Filer command prompt but then it does not accept any input.**

**5/18/2023:**

* **On the SurfacePro**
  + **the drive labeled Old Studio no longer seems to work**
  + **The drive labeled F: is not H:**
  + **I still have the drive returned from Dell. It probably needs to be reformatted.**
  + **The drive is not visible in the “Manage” app. I do not think that there is any way to reformat the drive.**
* **Moving back to use of CRTWindow---**
  + **Cannot print the event log (messages tab)**
  + **Not auto adjusting column widths-- can do logging to a file**
* **Value Range Error, Seg SCREENOP P#30, O#49. This is the call to UWRITE (2nd pass through SC\_Control). DbgCnt = 39979**
* **Should SetFunctionPrefixes be a method in TCRTInfo?**
* **TCRTInfoPsys.VersionNr is not being used**
* **SetFunctionPrefixes is never called**
* **SetFunctionPrefixes is defined in BOTH CRTUnit and in CRTWindow AND in MiscInfoUnit???**
* **Maybe it should be defined in CRTWindow and overridden in pSysWindow?**
* **Is CRTUnit unit a pSystem (specific) unit or a CRT unit (more generic)? Because CRTInfoPsys references TVersionNr, this is really a p-System unit.**

**5/19/2023:**

* **OnSaveSettings not defined?**
* **DeleteCh?**
* **Cursor keys have stopped working? Displaying the “ShortCuts” disabled the arrow keys-- even though the menu items were disabled.**
* **Save/restore position**
* **Key To Move to Prev Word (deleted)**
* **The date is still getting mangled!**

**5/20/2023:**

* **The wrong version of SetFunctionPrefixes is called for version I.4 (maybe for all p-System usage)**
  + **It gets created in DefaultCRTInfo**
  + **and in DefaultKeyInfo**
  + **These should be overridden in pSysWindow.Create**
  + **Should be using the other version of TCRTInfo: TCRTInfoPsys**
  + **The CrtWindow and PSysWindow ought to be calling their own private procedures to create the CRTInfo and KeyInfo**
  + **CRTInfo belongs to CrtWindow**
* **SetFunctionPrefixes is now a virtual procedure in TCRTWindows**
* **Must be called after LoadCrtInfo and after LoadCrtKeyInfo**
* **CRTInfo, KeyInfo must be created *before* CRTWindow or KeyInfo?**
* **CRTInfo is used for things like MAXCOLS, MAXROWS**
* **CrtInfoChanged is called by TCrtInfo.InfoChanged (but it was already called by ??)**
* **SetFunctionParameters should not need to have The\_LOw\_Func and The\_High\_Func passed in since they are part of CRTInfo.**
* **KeyToMoveCursorLeft is wrong**
* **The Debugger DisplayCrtKeyInfo is not setting the Prefixed info for the CRTInfo or KeyInfo that it creates.**
* **CrtKeyInfoStatusProc is displaying the CRTInfo before the prefixes have been set**
* **LookForUnusedFuncs is displaying stuff prematurely**
* **Previously SetFunctionPrefixes would set the prefixes as the final act of LoadKeyInfo or LoadCrtInfo. Maybe SetFunctionPrefixes could be passed as a parameter to LoadCtInfo or LoadKeyInfo**

**5/22/3023:**

* **SetFunctionPrefixes should really be a method of TCRTInfo**
* **fFunctionPrefixes is not set. SetFunctionPrefixes got called but it isn’t setting fFunctionPrefixes .**
* **“Key to End File” is the editor accept key in Version I.4**
* **What is the <etx> key supposed to do.**
* **Key Info screen isn’t working properly.**
* **DebugInterpreter is not disabling ^F**
  + **EnableMenus is called by BootInterpreter. It is not called by DebugInterpreter**
* **I am getting a stack overflow in version I.5 (LB) by just trying to go into the Filer (FILER.EXE not FILER\_DEBUGGER.EXE) (remember that MemDumpDF is not available in this version). DbgCnt = 35830. (Fetch->CXP->BLD3->STKOVR)**
* **When I try to boot F:\NDAS-I\d7\Projects\pSystem\Z80EM2010\UCSDII0.RAW, I get and AV in cbUnitNumberChange**
* **No argument for %s**
* **%s in the debugger settings**
* **Unknown ID:D DS in DisplayWatches**

| Peter Miller | V1.4 | PM V1.4 | UCSDI4 | F:\NDAS-I\d7\Projects\pSystem\Z80EM2010\UCSDI4.RAW | FILER.EXE | SetWordAt pass an ODD address |
| --- | --- | --- | --- | --- | --- | --- |

**5/25/2023:**

* **What is the editor “Accept” key for version I.4? It is ^Z. A.K.A. “Key to EndFile”.**
* **[F:\NDAS-I\d7\Projects\pSystem\Z80EM2010\UCSDI4.RAW] I am getting a “Range Check Error (ERangeError): Segment:0, ProcNum:17; RelIPC:10583” when I try to compile “HelloWorld” in version I.4 in the Peter Miller (FILER.EXE) followed by an AC at $0054df33. Compiles and runs fine when running FILER\_DEBUGGER.EXE??**
* **Version I.4 - none of the editor movement control keys work.**
* **Getting a stack overflow when I try to edit “HELLOWORLD” in version I.4 using FILER.EXE. This is the LB interpreter-- not the PM interpreter!**
* **The version with the debugger starts out with HL=$C25E**
* **The version w/o the debugger starts out with HL=$C25E**
* **Where is VISDIFF.EXE located?**
* **I am getting different results for HL in the call to SRO between FILER.EXE and FILER\_DEBUGGER.EXE**
* **Even though I am running the SYSTEM.VOL boot, LOGRUNS is showing UCSDI4.RAW. I did not**
* **$9EB0 (which is the munged up version of HL) comes from GLOBALS.LOWMEM.SYSCOM.LASTMP.**
* **Where does GLOBALS.LOWMEM.SYSCOM.LASTMP get set to $9EB2? It comes from SP in BLD3. (CBP→BLDMSCW→BLD3). The SP gets set to 40626 ($9EB2) in BLD3 when LASTMP is pushed onto the stack. Why is this different when running with the debugger?**

**5/26/2023:**

* **No debugger -vs- debugger**
* **When GETSEG is read in, the result (NEWSEGTOP) is different ($C282 -vs- $FC64)**
* **Also SegWasReadIn is different (TRUE -vs- FALSE)**
* **The debugger version looks the segment up to see if it is already in memory.**
* **There was code in GETSEG that was commented out when not compiling with “IfDef debugging”**
* **Trying to use ^D to move the cursor in version I.4 causes a beep which appears to come from the OS. Unit number 1.**

***4/14/202:***

* **Here is how I got into the p-System (on Debian):**
  + **~/bin/ucsdpsys\_vm -w ~/Downloads/system.vol**
  + **password for Debian is**
* **I get a stack underflow whenever I try to run UCSDI4.VOL or UCSDI5.VOL under the Debian interpreter.**
* **UCSDI4.RAW also does not accept any of the movement characters (UP ^E, LEFT ^S, etc)**
* **UCSDI5.VOL does seems to accept and properly handle all of the character movement characters.**

**5/27/2023:**

* **Trying to compile the E.4 editor using the Peter Miller I.4 system crashes**
* **Compiled without problem using the LB I.4 system. Getting an ODD address error when I try to edit.**
* **When I use the FILER to resize the volume, it does not seem to be recognized in the I.4 system. Cannot reproduce anymore.**
* **All of the files on the “SOURCE” volume are listed as “ILLEGAL” instead of “TEXTFILE”. Running FilerMain/Utilities/Misc/Clean changed the “ILLEGAL” to TEXTFILE”. Volume size is still wrong.**
* **Trying to compile again gives an ODD address error.**
* **the source code to the e.4 editor is on the SOURCE volume (aka E4EDITO.VOL)**
* **The arrow keys do work on the E.6 editor (Version I.5) when running the I.5 version. The E.6 editor does NOT work on the I.4 system.**
* **I get a range check error when trying to compile the E.4 editor on the PM I.4 system. The error is occurring in the *jump* function. Also getting an error when compiling under the PM I.5 system (occurring in GetWordAt ODD address: 000D, Segment:0, ProcNum:17, RelIPC:5). Getting an error in STB when I try to compile the editor using the PM 2.0 version (RCE (ERangeError; Segment:0, ProcNum:16; RelIPC:4) but it does finish the compile and does not report any errors from the p-System.**

**5/30/2023:**

* **The Z80 I.4 system compiles the E.4 editor without error**
* **I just compiled the E.4 editor using the LB I.4 FILER\_DEBUGGER.EXE (UCDI4.VOL) without error?**
* **I just compiled the E.4 editor using the LB I.4 FILER.EXE (UCDI4.VOL) without error?**
* **Trying to compile the E.4 editor using the LB I.5 FILER.EXE leads to a “stack overflow”. SP stats out around $C25E and bounces around, slowly declining.**
* **Trying to compile the F.4 editor using the PM I.5.FILE.EXE leads almost immediately to an “odd address” error in GETWORDAT**
* **I was forced to uninstall / reinstall Delphi because of numerous crashes when open Delphi or my project.**
* **I was able to Boot / Halt the following (I did not try to do anything after booting):**
  + **UCSDI4.VOL, I.4, Peter Miller, FILER\_DEBUGGER.EXE**
  + **UCSDI5.VOL, I.5, Laurence Boshell, FILER\_DEBUGGER.EXE**
  + **UCSDII.VOL, II, Peter Miller, FILER\_DEBUGGER.EXE**
  + **PSYSTEM.VOL, IV.22, Laurence Boshell, FILER\_DEBUGGER.EXE**
* **I was able to Boot / Halt the following** 
  + **PSYSTEM.VOL, IV.22, Laurence Boshell, FILER.EXE**
    - **I was able to edit/compile the E.4 editor**

**5/31/2023:**

* **I get a range check error when trying to compile the E.4 editor on the PM I.4 system. The error is occurring in the *jump* function.**
  + **F:\NDAS-I\d7\Projects\pSystem\Z80EM2010\UCSDI4.RAW**
  + **The JTab value has made a big change from $F3B6 to $CD6C**
* **Using the debugger with V1.4 has all sorts of inconsistencies:**
  + **OpCode Name does not match the decoded p-Code**
  + **DbgCnt does not match**
  + **AbsIPC does not match**
  + **RelIPC does not match**
* **The debugger crashes when it tries to UpdateDebuggerDisplay**
* **'Bytes @ $FCA5 [64677]: 00 9F CC 36 CD 05 01 C2 2A CA 36 9F CB A1 03 CD 05 01 CA 36 9F C3 A1 F6 D6 12 00 94 01 00 00 07 00 20 00 01 FF 92 0D 74 14 B8 16 C0 15 DE 19 54 1F 32'**
* **The I.4 p-Code window looks OK**
* **The I.4 watch line for wt\_OpCodesDecoded looks ok**
* **The version I.5 decoder is generating an “Unimplemented” message**
* **The I.5 decoder is handling the inspection window correctly**
* **The value passed in in Addr in TpCodeDecoderII.Decode never gets used**
* **Microsoft is screwing up Delphi’s settings: Losing things like Tools menu, etc. Maybe this is a result of re-installing Delphi? I have moved everything that I can off of the Desktop and into a (hopefully) private folder: “C:\My Private Desktop\”**
* **My descent into hell:**
  + **Trying to figure out why PM VI.4 gets a range check error when trying to compile OPSYS:EDITOR.TEXT**
    - **I want to be able to use the pCodeDebugger to trace through the code where the RCE is occurring.**
    - **MemDump not working for wt\_OpcodesDecoded -- completely ignoring the first (Addr) parameter. I.e. these are not consistent:**
      * **MemDumpDF(AbsIPC-1, 'B')**
      * **MemDumpDF(AbsIPC-1, 'O')**
    - **Compiler tools has gotten screwed up**
      * **VISDIFF.exe**
      * **UEDIT32.EXE**

**6/1/2023:**

* **fpCodeDecoder.OnGetByteBased is not assigned**
* **fpCodeDecoder.OnGetBaseAddress is not assigned**
* **BaseAddress (the last parameter to Decode) needs to be assigned**
* **Or can I assign “GetByteFromMemoryBased” to TDecodeToMemDump?**
* **Order:**
  + **Interpreter calls DecodedRange(0, 50)**
  + **which creates TDecodeToMemDump**
    - **TDecodeToMemDump needs to have the following assigned:**
      * **fOnGetByte2**
      * **fOnGetWord2**
      * **fGetBaseAddressFunc**
      * **fGetByteFromMemoryBasedFunc**
* **When V4 calls Bytes[p], p = 0. It should be calling BytesFromMemoryBased**
* **DUMMY1 := '''Bytes @ $F2B7 [62135]: 00 90 02 96 00 01 00 00 00 12 49 2F 4F 20 45 72 72 6F 72 20 6C 6F 61 64 69 6E 67 20 20 09 53 65 67 6D 65 6E 74 20 22 23 22 20 6E 6F 74 20 65 78 65 63''**
* **V4**
  + **Direct call MemDumpDF(AbsIPC-1, 'O') seems OK.**
  + **External windows decoder seems ok**
* **The following methods are ALL assigned to this fpCodeDecoder in pCodeDecoder:**
  + **OnAddLine := AddLine;**
  + **OnAddLineSeparator := AddLineSeperator;**
  + **OnGetByte3 := GetByteFromMemory;**
  + **OnGetWord3 := GetWordFromMemory;**
  + **OnGetJTAB := GetJTAB;**
  + **OnGetBaseAddress := GetBaseAddress;**
  + **OnGetCPOffset := GetCPOffset;**
  + **OnGetSegmentBase := GetSegmentBaseAddress;**
* **Where does GetByteFromMemory get assigned?**
* **Version II:**
  + **DUMMY1: 'Bytes @ $FBE0 [64480]: 9F CC 36 CD 05 01 C2 2A CA 36 9F CB A1 03 CD 05 01 CA 36 9F C3 A1 F6 D6 12 00 94 01 00 00 07 00 20 00 01 FF 92 0D 74 14 B8 16 C0 15 DE 19 54 1F 32 00'**
* **DecodeToMemDump needs to have GetByteFromMemoryBased assigned**
* **handlejb in pCodeDecoderII always evaluates to FALSE;**
* **TpCodeDecoderII.GetByteAt calls fGetByteFromMemoryBased(BaseAddress,fIPC) but fGetByteFromMemoryBased is not assigned.**
  + **Version IV dos NOT call GetByteAT**
  + **I reverted to the inherited version of GetByteAt**
* **The version I.4 decode window goes off into never never land (SLDC 0, SLDC0, …). Gets caught because the memory is all 0 and no RETURN code is ever encountered.**
* **But the watch window seems like it might be OK**

**6/2/2023:**

* **DecodeMemory1Click has the same value for SegBase, ProcBase**
  + **In versions I.4, I.5 SegBase is really the segment top -- not the base.**
* **Created a new function SegBot0 (used only in DecodeWindow.pas). This fixed the external decoder window from guessing a bad address for where to start the dump.**
* **i’m getting the same RCE even when I try to compile a trivial program in PM VI.4**

**6/5/2023:**

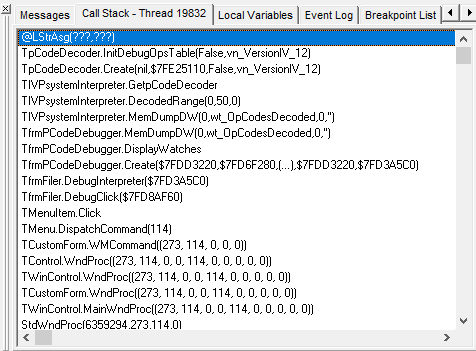
* **Here is how I got back into the Debian version of the p-system:**
  + **bin/ucsdpsys\_vm -w ~/Downloads/system.vol -w ~/Downloads/utility.vol**
* **I need to put a copy of “E4EDITO.VOL”l into the “public” volume on Delphi. Done.**
* **Can I boot from 15SYS1.VOL on Debian?**
* **Can Version II (“system.vol”) compile the EDITOR (located on E4EDITO.VOL?**
* **I need to use chown to change the ownership on both volumes.**
* **Chrome Remote Desktop Directly to Debian? NO. Too many problems. Back screen on remote user.**
* **Now, when I try to restart Debian, it tries to log me in multiple times.**
* **Do I have a backup copy of the Debian HD? Yes. It was saved to F:\Maxtor\Virtual Machines\linux\Debian 64A\_1 - Copy.vdi. I restored from the copy (everything changed since sometime in 2022 -- about February - was lost). Do I have a later copy somewhere?**
  + **I have “Debian Hard Disk - Copy.vhd” from the “Old Studio” backup “HD BU Hard Disk Backup 2” dates 1/31/2022**
  + **I also have another backup (on the same external HD) “H:\WD Elements\Maxtor\Virtual Machines\linux\Debian 64A\_1.vdi” dated 2/18/2022.**
  + **I have made a new backup of “Debian 64a\_1.vdi” which is on the “SurfacePro: WD Elements 2 (F:) Maxtor\Virtual Machines\linux\Debian 64A\_1 - Copy.vdi”**
  + **When I restored the SurfacePro would it have had another copy? No. Probably not since I only restored the C: drive.**

**6/6/2023**

* **I enabled “App & Browser Control” in Windows settings. Be alert for slowdowns or problems.**
* **I was able to get into the Debian p-System using the purple note above**
* **Booting attempts on Debian:**
  + **bin/ucsdpsys\_vm -w ~/public/UCSDI4.VOL → “Stack Underflow”**
  + **bin/ucsdpsys\_vm -w ~/public/UCSDI5.VOL → “Stack Underflow”**
  + **bin/ucsdpsys\_vm -w ~/public/UCSDII.VOL → “Stack Underflow”**
  + **bin/ucsdpsys\_vm -w ~/public/system.vol → boots ok**
  + **E4 editor compiles neatly using the version II system.vol**
  + **But when I try to execute EDITOR.CODE, get “ucsdpsys\_vm/main.c:511: CspLoadSegment: Assertion `!(SegSize & 1)' failed” from the newly compiled program.**
* **Now I should try to compile with the Delphi II system booting “system.vol”**
* **I can boot “system.vol” (the Debian version II boot disk) although it is very slow. Compiling is also very slow. Do I have any debug writeln code that might still be active?**
* **I get a “Range Check Error” on the line “ MemWrByte(Pop() { base address }, Offset, p1);” when the compile finished. It claimed 1698 lines/min.**
* **Yes. There were still a couple of MemDump lines active in the FETCH loop.**
* **Is the UJP working properly in the Version II system?**
* **After removing the extraneous MemDump lines, the compile claims to be 2185 lines / minute.After compiling with {$IfDef debugging} disabled the compile claims to be 17804 lines / min (still getting a RCE on the STB).**
  + **Cured the RCE by changing the STB op to: “p1 := Pop() and $FF”**
* **Running with the Version II system (“system.vol”) (i.e., “word\_Memory”) the “jump” instruction (see above) executes fine.**
* **Running with the Version I.4 system, the “jump” opcode executes 675 times before getting the RCE.**
* **By adapting the “jump” opcode in INTERPC to the code used in the “UJP opcode in the INTERPII code, it seems to work ok and executes at 27378 lines/min.**
* **But EDITOR.CODE won’t execute: “#5:editor.code not code”. I may have been trying to execute the .CODE file compiled under version II rather than the file compiled with the version I.4 system**
* **I need to restore UCSDI4.raw to a previous version. Restored from the web site.**
* **Copy current p-System sources to a DVD**
* **Need a task to make a surface backup of the entire XPS-8930 systematically.**
* **Need a task to backup the Virtual Box volume systematically.**

**6/13/2023:**

* **SYSTEM4 crash during startup**

****

* + **Version 4.12 DebugOpsTable.CSPEND = 50 but**
  + **Version 4.12 fInterpreterOpsTable.CSPEND = 40**
  + **Version 4.12 is creating a version 4.2 DebugOpsTable**
  + **Would be nice to have a *SAVE* button on the LoadVersion dialog**
  + **I am trying to implement the SAVE button. Where does it get saved? It gets saved to FILER.INI.**
  + **GetBootParams returns NIL when I click on SAVE**
  + **Needs;**
  + **NewBootParams := FilerSettings.RecentBootsList.Add as TBootParams;**
  + **NewBootParams.Assign(fRecentBootParams);**
  + **Still not making it into the RecentBootsList**
  + **Should be Idx = 4 in FilerSettings.RecentBootsList**
  + **VolumeName is empty. It never gets set (except when reloading from the settings)**
  + **Getting this error when just trying to *Save* (rather than debug). No error when booting.**
  + **Try CSPEND = 38**
  + **CspTable[0] and [1] not assigned. It's OK.**
  + **I’m back to AV in InitDebugOpsTable**
  + **CSPEND is now set to 40. Table length set to 41.**
  + **fDebugOpsTable.CSPEND = 50?**
* **fixed (?) the AV**
* **Getting to DbgCnt 39284 before getting an ODD address error in '6: USERPROG.READSEGR'. History:**

**'39283=SIND0 (USERPROG.READSEGR @669),39282=IXA (USERPROG.READSEGR @667),39281=DECI (USERPROG.READSEGR @666),39280=SLDC1 (USERPROG.READSEGR @665),39279=INC (USERPROG.READSEGR @663),39278=LDO (USERPROG.READSEGR @660),39277=IXA (USERPROG.READSEGR @658),39276=DECI (USERPROG.READSEGR @657),39275=SLDC1 (USERPROG.READSEGR @656),39274=INC (USERPROG.READSEGR @654),39273=LDL (USERPROG.READSEGR @651),39272=SCXG1 (USERPROG.READSEGR @649),39271=SLDC2 (USERPROG.READSEGR @648),39270=SLDC0 (USERPROG.READSEGR @647),39269=IXA (USERPROG.READSEGR @645),39268=DECI (USERPROG.READSEGR @644),39267=INCI (USERPROG.READSEGR @643),39266=SIND0 (USERPROG.READSEGR @642),39265=LDL (USERPROG.READSEGR @639),39264=INC (USERPROG.READSEGR @637)'**

* **As before, the TOS has gotten munged by 659 in READSEGR @ DbgCnt = 39,279**

**6/14/2023:**

* **Makes 4 passes successfully. Last good pass was**
* **DbgCnt = 37,232**
* **Has just pushed an “array base” of $5847 (odd) onto the stack. Crashes on the following SINDO which tries to load from that address.**
* **The EVEC gets munged when READSEGR is trying to “Save the segment name” at IPC 950**
  + **KERNEL - OK**
* **Closing the DashBoard window calls Destroy;**
  + **which calls AddWindow to the WindowsList**
  + **bu somehow doing the inherited Destroy brings us back into the Dashboard Destroy again (maybe the Notification does it).**
* **Most of the windows listed in the WindowsList are the names of debug windows (like: 'MemDump(DS, 'b','Segment UserProg', 0)'**
* **‘p-Code debugger Dashboard 0’ never got added to the windows list?**
* **But it *seems* to have gotten added at position 113**
* **I don’t think that the DebugSettings ever got saved**
* **When the debugger settings are saved, the WindowsList has not yet been updated**

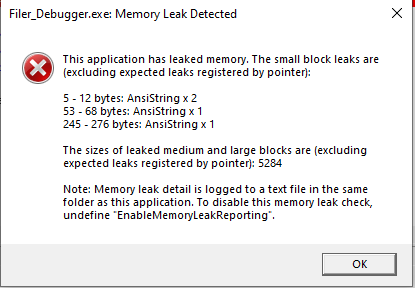
**6/16/2023**

* **[1] Trying to boot PSYSTEMY.VOL now says SYSTEM.MISCINFO is not found even though it is clearly on the volume.**
  + **Version appears to be [IV.2.2 R1.1]**
* **Searching for [Version]** 
  + **p1 does not seem to be advancing**
  + **Each loop should start AFTER the previous find**
  + **getting leaked memory**
  + **EUnhandledFlip leaking memory- fOnSVOLFree not assigned?**
  + **The memory leak does not occur when searching for an Ascii string even when the EUnhandledFlip exception is raised**
  + **The memory leak occurs even if my newly added code is commented out in pSysVolumes.ScanSegmentFile**
  + **The leak occurs even if I comment out the entirety of ScanSegmentFile**
  + **The volumes on which the error occurs, are left OPEN even when the error occurs**
  + **When I *delete* the problematic volumes (RL0.VOL, RLO-ORG.VOL, YTD.SVOL and TESTED.VOL) there is no memory leak.**
  + **I do not see RL0 getting mounted?**
  + **YTD.SVOL never gets freed: “Invalid Number of Files”**
  + **JUNK.SVOL never gets freed: “Invalid directory flip”**
* **Remember CharName function in MyUtils for use when displaying CRT control characters**

| **Versions found:** |  |
| --- | --- |
| **IV.13** | **F:\NDAS-I\d7\Projects\pSystem\Volumes\FORT\_SYS.VOL** |
| **IV.2.1 R3.3** | **F:\NDAS-I\d7\Projects\pSystem\Volumes\IV2UTILS.VOL** |
| **II.0** | **F:\NDAS-I\d7\Projects\pSystem\Volumes\Linux Vols\SYSTEM.VOL** |
| **IV.2.2 R1.1** | **F:\NDAS-I\d7\Projects\pSystem\Volumes\PSYSTEMX.VOL** |
| **IV.12 B** | **F:\NDAS-I\d7\Projects\pSystem\Volumes\SYSTEM4-SAVED.VOL** |
| **I.4b** | **F:\NDAS-I\d7\Projects\pSystem\Volumes\UCSDI4.VOL** |
| **I.5d** | **F:\NDAS-I\d7\Projects\pSystem\Volumes\UCSDI5.VOL** |

**6/17/2023:**

* **Is JUNK.SVOL being mounted BEFORE RL0.VOL?**
* **Yes. That is possible because JUNK.VOL and JUNK.SVOL come before RL0.VOL in the folder.**
* **I think that JUNK.SVOL is the culprit. No. I don’t think so.**
* **Where does *VolumeFile* get closed? It gets closed in *CloseVolumeFile*. *CloseVolumeFile* never seems to get called. I changed TVolume.Destroy to close the VolumeFile. That did not seem to change anything.**
* **The leak occurs even if JUNK.SVOL is not processed. It occurs even if ScanSegmentFile is completely commented out.**

****

* **Maybe the .SVOL is not getting freed?**
* **This finally came down to trying to continue processing a directory in ScanVolumeForVersionNumber even after EInvalidDirectory been raised.**
* **[1] Trying to boot PSYSTEMY.VOL. Getting a “Code Pool Overflow” error.**
* **Could there be a problem with Code Pool Info: Pool Outside; Pool Size: $7FFF; Pool Base: 0001:53f0; Resolution 16**
* **PSYSTEMY.vol boots after changing the Poolinfo to: Pool Outside. Pool size: 63; Pool Base: 0001:0000; Resolution: 16**
* **Back to trying to boot SYSTEM4.VOL**
  + **I changed the SYSCOM code pool info to Pool Outside. Pool size: 63; Pool Base: 0001:0000; Resolution: 16**
  + **and then tried to boot.**
  + **I get EOddAddress from GetWordAt passed an ODD address $5847, DbgCnt is 39284 - no change from before.**
  + **The code pool info from SYSTEM4A.VOL is Pool Outside. Pool size: 32767; Pool Base: 0001:5410; Resolution: 16**
  + **I changed SYSTEM4.VOL back to the original value: Pool Outside. Pool size: 32767; Pool Base: 0001:5410; Resolution: 16**
* **{here}**