Disassemble Memory File

Memory.DMP can be used to determine call graphs, outside the scope of analysis tools like kd.exe. With a full disassembly, the context around a particular symbol is explored at convenience.

```
uf nt!HalGetBusDataByOffset
├─────────────────────────▷uf nt!HalpGetPCIData
│                                               uf nt!HalpReadPCIConfig
│                                               uf nt!memcpy
│                                               uf nt!_security_check_cookie
└─────────────────────────▷uf nt!HalpGetSetCmosData
                                                   uf nt!HalpAcquireCmosSpinLock
                                                   uf nt!guard_dispatch_icall (N/A)
```

UfSymbol.ps1 renders the call graph based on a disassembly file. The file is generated once, reused at rendering stage. The disassembly is separated into individual function bodies. The root body contains the symbol requested by the user. A dependency graph contains the callers or the callees for each function. CLI switches determine the depth of the tree, target OS for rendering.

- Generic functions have many callers; ie. 1118 matches for `nt!KeBugCheckEx` at `-Depth 1`.

To keep the graph uncluttered, known functions are not disassembled: `KeYieldProcessorEx` calls other functions that are minute, `IofCompleteRequest`.

Sample output renders the call tree for `nt!KiSystemStartup`.

```
PS > (Measure-Command {
         $Image = 'D:\DataLake\2025-04-28\MEMORY.DMP'
         & '.\UfSymbol.ps1' -Symbol nt!KiSystemStartup -Image $Image -Depth 4 -Down | Out-Default
      }).TotalSeconds
File "D:\DataLake\2025-01-28\MEMORY.DMP" of 1194.36 Mb has been processed in 4570 seconds.
D:\Processing\53c6f2af-38db-4219-9f41-f794c7897f5a\53c6f2af-38db-4219-9f41-f794c7897f5a.disassembly
D:\Processing\53c6f2af-38db-4219-9f41-f794c7897f5a\53c6f2af-38db-4219-9f41-f794c7897f5a.meta
D:\Processing\53c6f2af-38db-4219-9f41-f794c7897f5a\53c6f2af-38db-4219-9f41-f794c7897f5a.retpoline
```

The 1st line shows an estimate for the minimum disassembly duration: a smaller file was processed in 1.26 hours on the same system. The decompilation is executed on all cores but 1. Besides the `.disassembly` file, `.meta` and `.retpoline` are created. The `.meta` file contains:

- *OS* and *computer* where the BSOD occurred
- *image* path and *hash*. The hash identifies duplicates, resulting in a decompilation bypass.
- *system* where disassembly took place, number of *cpus* alloted, cpu *model*, *duration* and image *size*.
- The default modules used to disassemble the image:
  - for a `.dmp` file *nt, pci, acpi and hal* functions are disassembled
  - *base name* for all others

The `.retpoline` file is an indirection table for bodies compiled with `/guard:cf`. Wherever `call nt!guard_dispatch_icall` is found, the source pointer is resolved in the memory file and displayed.

For `nt!KiSystemStartup` call tree:

- 1302 callees are identified with `-Depth 4`, 5318 at depth 6.
- Complete decompilation and identification takes **5215** seconds on an “Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz” with 3 cpus.

```
uf nt!KdInitSystem
├────────────────▷uf nt!KeQueryPerformanceCounter
│                                                uf nt!HalpTimerGetInternalData
│                                                uf nt!HalpTimerScaleCounter
│                 uf nt!ExAllocatePool2
│                 uf nt!_security_check_cookie
│                 uf nt!MmGetPagedPoolCommitPointer
├────────────────▷uf nt!KdRegisterDebuggerDataBlock
 …
uf nt!PpmUpdatePerformanceFeedback
uf nt!guard_dispatch_icall (nt!_security_cookie
                            nt!HalpOriginalPerformanceCounter
                            nt!HalPrivateDispatchTable+0x1b0=nt!HalpProcessorPrepareForIdle
                            nt!HalPrivateDispatchTable+0x1c0=nt!HalpProcessorResumeFromIdle
                            nt!HalpTimerReferencePage
                            nt!HalPrivateDispatchTable+0x418=nt!HalpLbrResumeRecording
                            nt!HalPrivateDispatchTable+0x2f8=nt!HalpTimerClockStop
                            nt!PopCsConsumption+0x140)
5215.506918
```

`-Setup` is a text based guide that configures the directory where disassemblies are stored. A symbol path can be specified, a lower limit can trigger a warning if other dissasemblies overlast it. Disassembly duration and system, cpu model, file size can be suppressed from future `.meta` files.

Notes

- Decompilation-ready processing is useful in support cases where the *Memory.DMP* file cannot be provided. Implementation differences between OS versions are also visible.
  - A `.dmp` file contains the dependencies from all modules, can trip the decompiler with inappropriate function bodies. This shortcoming does not apply to user mode.
  - An executable solves all functions, cannot solve dependencies.
- PowerShell *Core* is required. *Desktop 5.1* is slow.
- Hotpaths are moved to inflight *CSharp* assembly. Decompilation can be **8 times** faster.
- Decompilation through *kd.exe* can be superseded by *dbgeng.dll* COM interfaces.
- *UfSymbol* is meant for USB migration. No internet connection is needed.
- Where `(N/A)` appears in rendering:
  - indirection table has no corresponding target symbol - ie. register is used.
  - function is missing the body either due to absent module, or a large body has been decompiled and trimmed.
- `.retpoline` build is not parallelized. Only 2E+3 *poi* sources have to be decoded.
- Initially, the objective was GUI rendering through SVG. With broad trees being prevalent, a point-and-click is deemed impractical.

```
PS > $prefix = "https://raw.githubusercontent.com/armaber/scripts/refs/heads/disasm/";
     "functions.ps1", "UfSymbol.ps1" | foreach {
         Invoke-WebRequest $prefix/DisassembleImage/$PSItem -OutFile $PSItem;
     }
     Get-Help .\UfSymbol.ps1 -Full;
```