Disassemble Memory File

Memory.DMP used in post-mortem debugging can be processed without "Debugging Tools for Windows", particularly to obtain a call tree for a given function.

```
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 operates by storing the disassembly on a local database. The 1st time it is invoked, a parallel decompilation of the image takes place.

The disassembly is separated into individual function bodies. The root body contains the symbol requested by the user. A dependency graph is built either upstream, representing all the callers of the function, or downstream representing the callees. Care must be taken when specifying `-Depth`:

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

`$StopDisassembly` is a symbol table where parsing stops: `KeYieldProcessorEx` calls other functions that are minute, `memset`, `atoi`, `KeStallExecutionProcessor`, `IofCompleteRequest` are not explored.

Sample output builds 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 gives a heads-up about the disassembly duration: a smaller file was processed in 1.26 hours on the same system.

The decompilation is done in parallel using all cores but 1. Once completed, 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 function pointer is resolved in the memory file and displayed.

For `nt!KiSystemStartup` call tree:

- 1302 callees are identified with `-Depth 4`
- Complete decompilation and identification took **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
```

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.
- PowerShell *Core* is required. *Desktop 5.1* is slow.
- `.retpoline` built is not parallelized.
- SVG rendering is not implemented.
- *UfSymbol* is meant for USB migration. No internet connection is necessary.
- Removing CR character from the large disassembly can result in *OutOfMemory* exception.
- Some symbols might not be identified if the image is a *Memory.DMP* file. Decompilation hits `uint16_t` zeroed memory which coincides with an opcode. The runspace is not killed by the *OutOfMemory* exception; other side effects can occur. This shortcoming is being addressed.

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