TEB + 0x800 is Win32ClientInfo structure

```
kd> dq @$teb+800 L6
000000b1`1f1c8800  00000000`00000008 00000000`00000000
000000b1`1f1c8810  00000000`00000600 00000000`00000000
000000b1`1f1c8820  0000028b`98270700 0000028b`98270000
```

Offset 0x20 used to be a pointer to a tagDESKTOPINFO structure and offset 0x28 used to be the ulClientDelta. Inspecting the content of the new buffer at offset 0x28 shows:

```
kd> dq 0000028b`98270000
0000028b`98270000  00000000`00000000 01006db0`8996dbe9
0000028b`98270010  00000001`ffeeffee ffffcf00`00800120
0000028b`98270020  ffffcf00`00800120 ffffcf00`00800000
0000028b`98270030  ffffcf00`00800000 00000000`00001400
0000028b`98270040  ffffcf00`008006f0 ffffcf00`01c00000
0000028b`98270050  00000001`000011f8 00000000`00000000
0000028b`98270060  ffffcf00`00a07fe0 ffffcf00`00a07fe0
```

This contains kernel address in lots of places and is actually the user mode mapped version of the kernel desktop heap, which in this case is at 0xFFFFCF0000800000. Inspecting that address reveals the same content:

```
kd> dq ffffcf00`00800000
ffffcf00`00800000  00000000`00000000 01006db0`8996dbe9
ffffcf00`00800010  00000001`ffeeffee ffffcf00`00800120
ffffcf00`00800020  ffffcf00`00800120 ffffcf00`00800000
ffffcf00`00800030  ffffcf00`00800000 00000000`00001400
ffffcf00`00800040  ffffcf00`008006f0 ffffcf00`01c00000
ffffcf00`00800050  00000001`000011f8 00000000`00000000
ffffcf00`00800060  ffffcf00`00a07fe0 ffffcf00`00a07fe0
```

While the UserHandleTable is gone, it is possible to just search the pure data looking for an object handle and adding the offset into the user mode mapping to the base of the kernel desktop heap to gain the true kernel address.