

```

*****
23258 Mon Apr  9 21:31:15 2018
new/usr/src/cmd/mdb/i86pc/modules/unix/i86mmu.c
9452 ptable_dcmd() needs a little cleanup
*****
_____unchanged_portion_omitted_____

826 static int
827 do_ptable_dcmd(pfn_t pfn, uint64_t level)
828 {
829     struct hat *hatp;
830     struct hat hat;
831     htable_t *ht;
832     htable_t htable;
833     uintptr_t base;
834     int h;
835     int entry;
836     uintptr_t pagesize;
837     x86pte_t pte;
838     x86pte_t buf;
839     physaddr_t paddr;
840     size_t len;

841     /*
842      * The hats are kept in a list with khat at the head.
843      */
844     for (hatp = khat; hatp != NULL; hatp = hat.hat_next) {
845         /*
846          * read the hat and its hash table
847          */
848         if (mdb_vread(&hat, sizeof (hat), (uintptr_t)hatp) == -1) {
849             mdb_warn("Couldn't read struct hat\n");
850             return (DCMD_ERR);
851         }

852         /*
853          * read the htable hashtable
854          */
855         paddr = 0;
856         for (h = 0; h < hat.hat_num_hash; ++h) {
857             if (mdb_vread(&ht, sizeof (htable_t *),
858                 (uintptr_t)(hat.hat_ht_hash + h)) == -1) {
859                 mdb_warn("Couldn't read htable\n");
860                 return (DCMD_ERR);
861             }
862             for (; ht != NULL; ht = htable.ht_next) {
863                 if (mdb_vread(&htable, sizeof (htable_t),
864                     (uintptr_t)ht) == -1) {
865                     mdb_warn("Couldn't read htable\n");
866                     return (DCMD_ERR);
867                 }
868             }

869             /*
870              * Is this the PFN for this htable
871              */
872             if (htable.ht_pfn == pfn)
873                 goto found_it;
874         }
875     }

876     }

877 }

879 found_it:
880     if (htable.ht_pfn == pfn) {
881         mdb_printf("htable=%p\n", ht);
882         if (level == (uint64_t)-1) {
883             level = htable.ht_level;

```

```

884     } else if (htable.ht_level != level) {
885         mdb_warn("htable has level %d but forcing level %lu\n",
886             htable.ht_level, level);
887     }
888     base = htable.ht_vaddr;
889     pagesize = mmu.level_size[level];
890 } else {
891     if (level == (uint64_t)-1)
892         level = 0;
893     mdb_warn("couldn't find matching htable, using level=%lu, "
894         "base address=0x0\n", level);
895     base = 0;
896     pagesize = mmu.level_size[level];
897 }

899     paddr = mmu_ptob((physaddr_t)pfn);
900     for (entry = 0; entry < mmu.ptes_per_table; ++entry) {
901         len = mdb_pread(&pte, mmu.pte_size,
902             paddr + entry * mmu.pte_size);
903         if (len != mmu.pte_size)
904             return (DCMD_ERR);
905         pte = buf;

906         if (pte == 0)
907             continue;

909         mdb_printf("[%3d] va=0x%p ", entry,
910             VA_SIGN_EXTEND(base + entry * pagesize));
911         do_pte_dcmd(level, pte);
912     }

914 done:
915     return (DCMD_OK);
916 }
_____unchanged_portion_omitted_____

```