Lab 3 Task 1

Danny Topete
CS 153, Prof. Song

May 23, 2025

- Explain how *walkphdir* defined in *vm.c* works. Specifically, what are *PDX*, *PTX*, *PTE_E*, *PTE_ADDR*, *PTE_W*, and *PTE_U*?
  - *walkphdir* is a function that locates the Page Table Entry (PTE) for a given Virtual Address (va) in a given page directory (pgdir). If the relevant page is not found, it allocates the memory for a new page table.
  - *PTE_E*, Checks if the page table is present
  - *PDX*, Extracts the page directory index
  - *PTX*, extracts the page table index
  - *PTE_P*, Page Table Entry Present, indicates if the page is present in memory
  - *PTE_W*, PTE Writable, indicates if page is writable
  - *PTE_U*, PTE User flag, indicates if page is accessible from user mode
  - *PTE_ADDR*, extracts the physical address of the page table entry

  1. Find Page Directory Entry,

     ```
             pde = &pgdir[PDX(va)];
     ```

  2. Check if Page Table Exists

     ```
     if(*pde & PTE_P){
       pgtab = (pte_t*)P2V(PTE_ADDR(*pde));
     }
     ```

  3. Allocate a New Page Table if needed

     ```
     else {
       if(!alloc || (pgtab = (pte_t*)kalloc()) == 0)
         return 0;
     ```

```
  memset(pgtab, 0, PGSIZE);
  *pde = V2P(pgtab) | PTE_P | PTE_W | PTE_U;
}
```

4. Return the Address of the PTE

   ```
   return &pgtab[PTX(va)];
   ```

- Reason about *structkmap* and *setupkvm*, explain how $P2V$ and $V2P$ work? Specifically, why add and subtract $KERNBASE$ can convert physical address to/from kernel virtual address?
- Based on the above understanding, explain how *uva2ka* maps user virtual address to kernel address.