XV6 Null Pointer Dereference in Two-Level Page Tables

1. **Project Objective:**
   - How XV6 initializes address spaces via exec()
   - How page tables are copied during fork()
   - The kernel's mechanism for handling page faults and dereferences a null pointer in the two-level page table of Xv6.
2. **Key Components Analyzed:**

A. XV6's Two-Level Page Table Structure

Code:

```
 // kernel/mmu.h   is location
 typedef uint64 pte_t;    // Page Directory (points to page tables)
typedef uint64 *pagetable_t; // 512 PTEs per page table   Page Table (maps to physical pages)
```

B. Critical Functions

Function	Location	Purpose
exec()	kernel/exec.c	Loads program into memory, sets up page tables
fork()	kernel/proc.c	Copies parent's page tables to child
walk()	kernel/vm.c	Translates VA to PA using page tables

3. **Build and implementation:**
   - ❖ **Prerequisites:**
     - RISC-V toolchain (riscv64-unknown-elf-gcc)
     - QEMU (qemu-system-riscv64)
   - ❖ Step:
     1) **Clone XV6**: git clone https://github.com/mit-pdos/xv6-riscv.git
     2) Modified and Overwrite kernel/trap.c and kernel/start.c

3) Add user/null_deref.c file
4) Modified Makefile add _null_deref\ inside uprogs from root directory of xv6
5) Run terminal command make clean & make qemu
6) Now xv6 os booted and open interface of xv6
7) Run xv6 terminal command null_deref
8) Expected output :

```
 Dereferencing null pointer...
 pid 3: page fault at 0x00000000
Null pointer detected!
```

Explore the full source code on GitHub: https://github.com/digontobiswas/XV6-Null-Pointer-Dereference-in-Two-Level-Page-Tables

-------End-------

Documented by:

Digonto Biswas
Roll-23053429
Section-CSE38
KIIT DU