Capítulo 1

Task 13: Update Child Node for Access

1.1 Task Description

The goal of this task is to modify the exception handler to dynamically map addresses during a page fault. Specifically, when a page fault occurs for a pointer (`ptr3`), the handler should map the address to the same physical page as `ptr1` and `ptr2`. Other entries in the page table should remain absent.

1.2 Planned Implementation

1. **Modify the Exception Handler:**
   - Update the `page_fault_handler` function to handle page faults for `ptr3`.
   - Dynamically map the faulting address to the same physical page as `ptr1` and `ptr2`.
2. **Set Up the Page Table:**
   - Ensure the root page table (`raiz`) has an entry pointing to a child node for the second-level page table.
   - Initialize the child node with all entries set to absent.
3. **Create Pointer `ptr3`:**
   - Define a pointer `ptr3` that points to a virtual address 4MB ahead of `ptr2`.
4. **Trigger a Page Fault:**
   - Attempt to read from `ptr3` to trigger a page fault.

5. **Map the Address:**
    - In the exception handler, map the faulting address to the same physical page as `ptr1` and `ptr2`.
6. **Verify the Mapping:**
    - Print the contents of `ptr1`, `ptr2`, and `ptr3` to confirm they all point to the same physical memory.

1.3 Expected Outcome

- The exception handler dynamically maps the faulting address for `ptr3`.
- The contents of `ptr1`, `ptr2`, and `ptr3` are identical, confirming they point to the same physical memory.
- Other entries in the page table remain absent.

1.4 Implementation Details

(To be filled after implementation)

1.5 Challenges

(To be filled after implementation)

1.6 Final Outcome

(To be filled after implementation)