Capítulo 1

Task 06: Remove Mapping

1.1 Objective

The goal of this task is to remove a memory mapping and observe the behavior when accessing unmapped memory. This involves:

- Reading the content of a pointer (`ptr2`).
- Removing the mapping associated with `ptr2`.
- Attempting to read the content of `ptr2` again.
- Documenting the expected and actual results.

1.2 Implementation Plan

1. **Setup:** Use the existing setup from the previous task where `ptr2` is mapped. Ensure `ptr2` points to a valid memory location initially.
2. **Remove Mapping:** Modify the page table to remove the mapping for `ptr2`. Use the `invlpg()` function to invalidate the TLB for the unmapped address.
3. **Access Unmapped Memory:** Attempt to read the content of `ptr2` after the mapping is removed. Observe and document the behavior.
4. **Expected Results:** Accessing `ptr2` after the mapping is removed should trigger a page fault. The system should handle the page fault gracefully if an exception handler is installed.

1.3 Expected Outcome

- The program should demonstrate the effect of removing a mapping on memory access.
- A page fault should occur when accessing `ptr2` after the mapping is removed.

1.4 Code Example

Below is a simplified example of how the mapping removal might be implemented:

```
void remove_mapping(void *ptr) {
    // Modify the page table to remove the mapping
    page_table[INDEX(ptr)] = 0;

    // Invalidate the TLB for the specific address
    invlpg(ptr);
}
```

Listing 1.1: Removing a Mapping

1.5 Results

(To be filled after implementation and testing.)

1.6 Challenges

(To be filled after implementation and testing.)

1.7 Conclusion

(To be filled after implementation and testing.)