Project 3 Report

CS436 - OS Design

Omer Sen: os226
Reuben Thomas: rmt135

1. Detailed logic of how you implemented each virtual memory function.

get_next_avail(): This returns the next available page in either virtual or physical memory using the bitmaps. If it's the virtual bitmap, it will also look for contiguous pages if multiple pages are being allocated

set_bit_at_index() and get_bit_at_index(): These set/get a given bit in a bitmap

set_physical_mem(): This function initializes the physical memory, determines the number of bits for the offset, inner, and outer page indexing, initializes the bitmaps, and initializes the TLB.

Translate(): This function first checks the TLB for a translation and returns it if it's there. Otherwise, it extracts the page table indices and offset and walks the page table. If it obtains a physical address, it stores it in the TLB and returns it. Otherwise it returns NULL.

page_map(): This function extracts the page table indices and offset and walks the page table. If it obtains a physical page number it returns it. Otherwise it checks if the top-level or second-level PTEs are uninitialized and initializes and allocates physical pages for them if so. It then adds the physical page allocated for the user to the TLB.

t_malloc(): This function allocates virtual page(s) for the user, and maps each page to a physical page using paeg_map(). It then returns the address of the first virtual page.

t_free(): This function frees the virtual page(s) given by the user and their respective physical pages. It also removes the entry from the TLB if there is one.

put_value(): This function copies data from a given buffer to the physical page given by the user. At the end of the physical page, the virtual address is retranslated to the new physical page.

get_value(): This function copies data from a physical page to a given buffer. At the end of the page, the virtual address is retranslated.

mat_mult(): This function gets and puts the appropriate data in order to multiply the two given matrices and store the result in the final given matrix.

add_TLB(): Adds a given translation to the TLB using a % hash function.

check_TLB(): Checks TLB for a given virtual page address. Also increments TLBchecks and TLBhits accordingly.

print_TLB_missrate(): prints the miss rate as a double using TLBhits and TLBchecks.

2. Support for different page sizes (in multiples of 8K).

The library supports page sizes that are multiple of 8KB. The address will be appropriately segmented, and the page tables will be appropriately allocated.

3. Possible issues in your code (if any).

None.

4. Collaboration and References: State clearly all people and external resources (including on the Internet) that you consulted. What was the nature of your collaboration or usage of these resources?

I consulted some pages online to get a refresher of the matrix multiplication algorithm.