Devin Macalalad dtm97
David Gasperini dlg195

1)
set_physical_mem: mallocs using MEMSIZE to allocate for physical mem, which is just a char (byte) array. Uses calloc to allocate and zero out bitmaps for physical and virtual pages. The global list of TLB structures are also allocated and set to default values.

add_TLB: right shifts the virtual address given by pgbits(offset bits) to be able to index to proper TLB structure given virtual address. If the TLB at the indexed slot is occupied, the occupying page is removed via remove_TLB. The valid is then set to true, the virtual address and physical address are then set to the passed arguments.

remove_TLB: right shifts the virtual address given by pgbits(offset bits) to be able to index to proper TLB structure given virtual address. Sets the validity to false, and both pointers to NULL.

check_TLB: TLB works by having an address map to a TLB structure by using a modulo on the virtual address passed in with the total number of TLB structs (index=va%TLB_ENTRIES). check_TLB right shifts the virtual address given by pgbits(offset bits) to be able to index to proper TLB structure given virtual address. If the valid member is set to false, the miss counter is incremented, and NULL is returned. Otherwise, the physical address in the TLB is returned. The counter of total lookups is incremented regardless.

print_TLB_missrate: Counts the number of misses and total lookups in check TLB, then performs some division to calculate the value.

translate: Has two basic steps: 1. Check the TLB and return the translation or 2. Do the translation and add it to TLB

page_map: Indexes to specific page directory entry using top bits. If the page table of the page directory entry is NULL, a page table is allocated using malloc, with each physical address being set to NULL. Then a specific page table entry is indexed to using middle bits. Finally, the physical address of that page table entry is set to what was passed.

get_next_avail: This function will attempt to find a streak of n repeating false (0) values using bitwise operations, and if it can it will return the index of the first bit. Due to much investigation, we determined the only reliable optimization for this is to skip a ‘11111111’ byte to avoid needlessly checking if the 1s equal 0. This was later modified to work for both physical and virtual space and modify the respective bitmaps.

a_malloc: If the memory it not initialized, then it will initialize. Then it will calculate the number of requested pages and call get_next_avail, then modify the return pointer as needed and return the first page pointer.

a_free: It will get the virtual to physical translation (by calling translate), then 0 those bitmaps. Free will also remove the translation from the TLB as it is now invalid, and zero the memory.

Mutex: both malloc and free use a pthread mutex to protect access to global structures.

put_value: number of pages to hold the passed size is calculated. A loop iterated through this number, each time finding the physical address mapped to the virtual address + the current page via translate. Then a memcpy is used to copy the data from val + the current page to the physical address. If the current page is the last needed, size%PGSIZE amount of data is copied instead of a whole page.

get_value: same as put_value except the current page of the physical address is being copied into val + the current page

mat_mult: Two nested loops count through the rows and columns, while a third inner loop is used to be able to address each matrix separately to perform a correct multiplication. Get_value is used to obtain the values from the two matrices. They are then multiplied together and added to the running sum for the current position using put_value.

2)
Benchmark Output:
Allocating three arrays of 400 bytes
Addresses of the allocations: 1, 1001, 2001
Storing integers to generate a SIZExSIZE matrix
Fetching matrix elements stored in the arrays
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
Performing matrix multiplication with itself!
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
Freeing the allocations!
Checking if allocations were freed!
free function works

TLB miss rate 0.005964

3) The code supports page sizes in multiples of 4K.

4)
Possible issues:

- Anywhere malloc/calloc is used, a failure will result in an error of the over all program