OS Project 3

Nick Belov: ndb68

April 11, 2024

1 Virtual Memory Functions

1.1 set_physical_mem()

In this function I initialize the "physical" memory and all auxiliary data structures. I place the page_table inside the physical memory and initialize two global bitsets for managing pages and memory. I also initialize a global mutex lock which I used throughout the rest of the program to ensure thread safety.

1.2 page_map()

First I actually calculate where the page_table is located in physical memory (this depends on the page size). Then I actually traverse the data structure.

Since its a two layer page_map I split the prefix of the virtual addresses into two sections and use that to traverse the tree. Exactly how the prefix is split depends on the page size but its all done nicely through macros so I as a programmer did not have to think about it.

After doing some bitwise magic to grab the proper indices I traverse the tree to check if a page exists. If it doesn't I grab the next available page with page_mex() and place it.

1.2.1 page_mex()

This just loops over the pages and checks them in the bitset until it finds an unused one.

1.3 page_map_tlb_wrapper()

I use this simple wrapper function to check the tlb for a page before calling page_map. It checks for a virtual memory address in the *tlb* and if it misses it adds it after calling page_map (overwriting anything already there).

1.4 translate()

Here after using page_map_tlb_wrapper() to get the pageid I just do some bitwise math to calculate the physical address. Again this all depends on the page size.

1.5 t_malloc() & t_free()

1.5.1 t_malloc()

Here I do a two-pointer on the virtmem bitset to find a contiguous section of virtual memory big enough. Then use page_map_tlb_wrapper() to make sure all the pages are mapped, updating the virtmem bitset in the process.

1.5.2 t_free()

Here I traverse the interval of virtual-memory, ensure its currently allocated with the vertmem bitset, and deallocate it (turn the bits off).

1.6 put_value() & get_value()

Both of these functions work the same way. I iterate over each bit I intend to copy, verify that the virtual memory is valid, then copy from the translated address or to the translated address.

1.7 add_TLB() & check_TLB()

1.7.1 add_TLB()

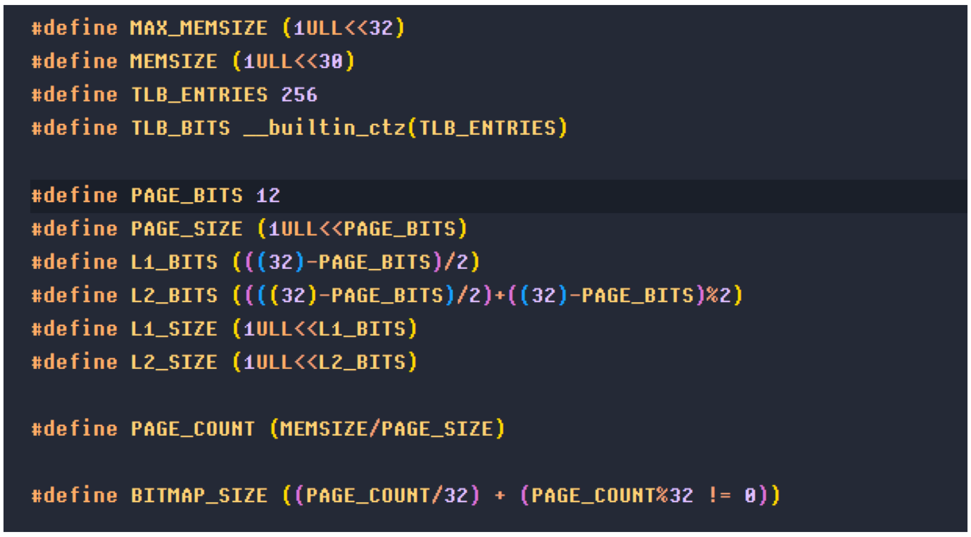
To determine the index in the TLB that a virtual memory address corresponds to I just take some fixed suffix of its bits (exactly how much depends on the size of the TLB). Then I overwrite whatever is currently there!

1.7.2 check_TLB()

The actually TLB is just an array with key, value pairs. I can determine what the index of a virtual memory address should be again by looking at the suffix of its bits, then I compare the virtual memory address to the key at that location. If they are equal its a hit.

2 Page Sizing

To implement dynamic page sizing I make extensive use of macros throughout the program that depend on the initial page size. Things like the exact structure of the page_table or the number of entries all depend on the page size.



```
#define MAX_MEMSIZE (1ULL<<32)
#define MEMSIZE (1ULL<<30)
#define TLB_ENTRIES 256
#define TLB_BITS __builtin_ctz(TLB_ENTRIES)

#define PAGE_BITS 12
#define PAGE_SIZE (1ULL<<PAGE_BITS)
#define L1_BITS (((32)-PAGE_BITS)/2)
#define L2_BITS ((((32)-PAGE_BITS)/2)+((32)-PAGE_BITS)%2)
#define L1_SIZE (1ULL<<L1_BITS)
#define L2_SIZE (1ULL<<L2_BITS)

#define PAGE_COUNT (MEMSIZE/PAGE_SIZE)

#define BITMAP_SIZE ((PAGE_COUNT/32) + (PAGE_COUNT%32 != 0))
```

3 Potential Issues & Testing

To test the program I malloced, populated, and freed a large array several times all while ensuring the values inside of it were correct. I also tried to multiply matrices and verified the correctness of the values. I did this all with multiple page sizes to ensure the abstractions worked nicely.

As for issues I think the actually strategy for allocating virtual memory is quite bad and could lead to slow allocation times and high fragmentation.

4 Collaboration and References

Worked by my self on this one! As for references I got a lot implementation advice from my friend Rajat and I used ChatGTP to help tinker with the Makefile.