UNIVERSITY OF COPENHAGEN

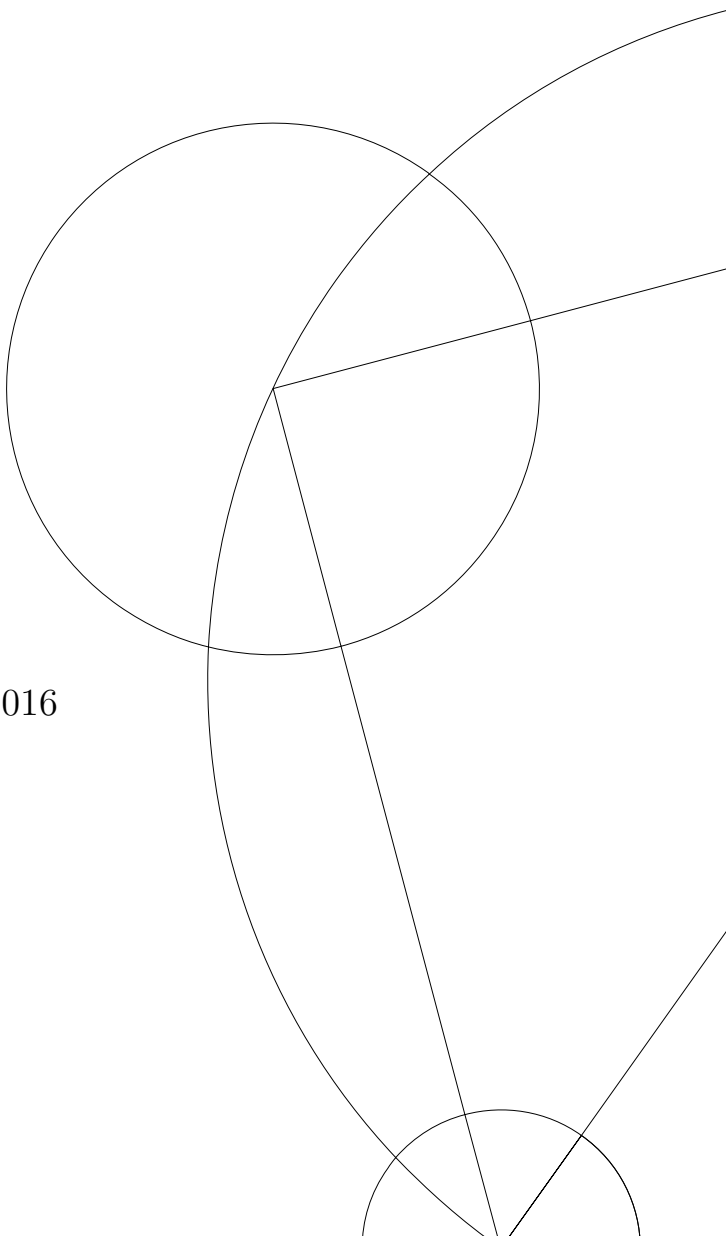
G4

Userland Semaphores and thread-safe priority queue

Mikkel Enevoldsen

Kristian Høi

Simon Van Beest



March 11, 2016

Task 1

Implementation

In handling TLB exceptions we have handled 3 cases; Load, store and store in case the page is not writeable. First and foremost we have in each case made use of the int 'mode' which indicates wether we are dealing with a user thread or a thread from the kernel. In all cases we end by triggering kernel panic if we get a thread from the kernel, and exits the process if it is a user thread.

In the last two cases we start with finding the page that matches the address. We use the 'find_matching_page' for this, which leads our pagetable through og returns the relevant index based on our thread entry.
We check if there is mapped to an even or uneven adress, which is used to check V0 or V1. Afterwards we check based on our valid bit, if there can be written. If this is the case we use a random replacement strategy.
Our exceptionhandlers are added, so that they are called when a TLB error occurs. Vores exceptionhandlers er tilføjet så de kaldes ved TLB exceptions.

Testing

Testing here is not straight forward. We expect the exceptionhandlers to be working, because we know that a TLB exception triggers our functions. We are also able to run our programs, which should invoke a TLB exception.

Task 2

Implementation

In implementing 'memlimit' we first check if the argument 'new_end' is NULL. If it is we return the 'old_end'. If the 'new_end' is lower than the 'old_end' we return NULL. If both these tests fail, we allocate memory for the physical page, maps the virtual memory to the physical page and then updates the heap end. Finally we return the new heap end.

Testing

The 'mem0.c' test seems to be working as expected, eventhough we doubt that the implementation om dynamic memory alocation is complete. This doubt comes from our own test 'testmem.c', which allocates memory, writes to that memory, updates the heap end, allocates more memory, writes to that memory and finally tries to print it to the terminal. This should work, but gives an error councerning re-mapping.