

Lab 1:

Objective: Developing a memory management system in C/C++ as per specifications given below.

Start date: August 13, 2024

Deadline: ~~August 26, 2024~~: Sept 2, 2023

Total points: ~~20~~ 30

Maximum team size: 2

Specifications:

Memory management system will have three components:

1. Memory manager: Function of memory manager is (a) to keep a record of available physical pages, (b) If a request comes, it returns the physical page address.
2. Task: Task module can have many instances. Each instance will have its own page table. Task will request memory manager for physical memory based on trace file provided.
3. Performance module: Computes total memory required for page tables, physical memory allocated to each task, available (free) physical memory.
 - a. Run trace containing 1000,000 lines, and report execution time for all the three implementation : map, single level page table and two level page table
 - b. Report total memory required for page tables, physical memory allocated to each task, available (free) physical memory.
 - c. Report number of page hits and misses for every process
4. Input/Output module: Will read a trace file, invoke a particular instance of "Task". Format of trace:
 - a. <Task>: <Logical address in hex>: <Size>Example:
T1: 0x4000:16KB
T1: 0x6000:4KB
T2: 0x40000:16KB
it can be assumed that tasks will start from capital T, Logical memory address will be hex. Memory size will be either in KB or MB.
There is a possibility of the address getting repeated. Repeated addresses are marked as page table hits.
5. Test module: Test module will generate traces. It takes a number of tasks as input. Starting address of text section, data section, stack section, shared library section, heap section is fixed in the config.h file. Address are randomly generated. All the addresses are aligned to minimum page size boundary, that is, if minimum page size is 1KB, all the local addresses are multiple of 1KB.

Features:

1. Page table will have multiple implementations
 - a. As a map. Logical page number is the key and physical page number is value
 - b. As single level page table
 - c. As multiple level page table, entries in each level is defined in config.h
2. Configurable parameters:
 - a. Page sizes
 - i. Multiple page size
 - b. Virtual memory size 2^N
 - c. Physical memory size 2^N
3. Test module will use pthreads to generate test cases. Each task will be assigned to one pthread.