**1.1 Requirements**

The Task as us to configure and run a memory management simulator. The commands and memory.conf we can edit them to set a correct configuration and observe the result in “trace file”. We expected to map 8 pages of physical memory to the first 8 pages of virtual memory, read from one virtual memory address on each of the 64 virtual pages, and predict which virtual memory addresses cause page faults. What page replacement algorithm is being used.

**1.2 memory.conf**

A screenshot of a cell phone

Description automatically generated

The page size is 16384.

* 1. **commands file**

We can find the page size 16384 as defined in memory.conf file, after mapping the memory, we have to write 64 valid memory address, we increased the address between the page length always by 16384.

**A close up of a logo

Description automatically generated**

* 1. **Result**

**A picture containing computer, keyboard

Description automatically generated**

* 1. **Observation**

It is clearly we can see that it is impossible to map only 8 and read 64 pages, cause the number of pages was always equal to the sum of physical and virtual pages mapped. If we set the maximum number of pages to 32, the simulator always mapped 16 physicals to 16 virtual pages, and if we set the maximum number of pages to 16, the simulator always mapped 8 physicals to 8 virtual pages. We can assume that by running the simulator that after page 31 the page faults will occur that means we are reading out of the range or in invalid direction of memory. And it may be an interruption caused by the memory management unit of the central processor when the software attempts to access a page that has been mapped in the virtual address space but is not currently loaded in physical memory. Although we have page faults, the virtual pages still allocated sequentially in the physical ones.

**The page replacement algorithm** is an algorithm decides which pages should be writing to disk when new page needs to allocate. The algorithm used in this case is “FIFO” (First-In First-Out). A FIFO replacement algorithm links with each page the time when that page was added into the memory; the oldest page is chosen when a page is going to be replaced. We can create a FIFO queue to hold all the pages present in the memory disk. At the head of the queue we replace the page. We insert page at the tail of the queue when a page is added into the memory disk.

**Disadvantage of FIFO algorithm** is low efficiency, because it doesn’t take under consideration which pages are frequently used and which are used once.