* All the phases - phase 1: Simple Paging, phase 2: Demand Paging, and phase 3: Demand paging with page replacement using ageing have been completed.
* In **paging**.c, Utility functions to find allocated memory, get pending page, get previous page and get next page have been implemented.  
  Also function to print gage fault frame is used in page fault handler function in order to print information about the frame that has been allocated before any changes are made (this includes getting to know if the page was free or we obtained the oldest frame).
* In **admin**.c a timer has been set up a that performs age scan.
* In **cpu**.c, code to handle page fault and periodical age scan for demand paging has been added. The code for instruction execution was added inside the execute\_instruction () function.
* In **process**.c, context\_in and context\_out functions have been written. In **simos**.h the CPU and PCB structures have been updated to include a new attribute for data offset. This was used in process.c loader.c and paging.c.  
  Code to implement idle quantum was added which limits the amount of time the idle process runs before an admin requires to give a new command.

Group member contributions:

NXG165630 Dinidu Gunasena: implementation of free list and writing back to swap disk in paging.c; loader functions in loader.c; implementation of swap queue, implementation of swap thread semaphores in swap.c.

RXC170010 Rajarshi Chattopadhyay: code for page fault handling and periodical age scan in cpu.c; code for executing instructions in cpu.c; functions for memory access and memory and memory frame management, page table management, memory age scan have been implemented in paging.c. Code to handle idle quantum, process execution in process.c