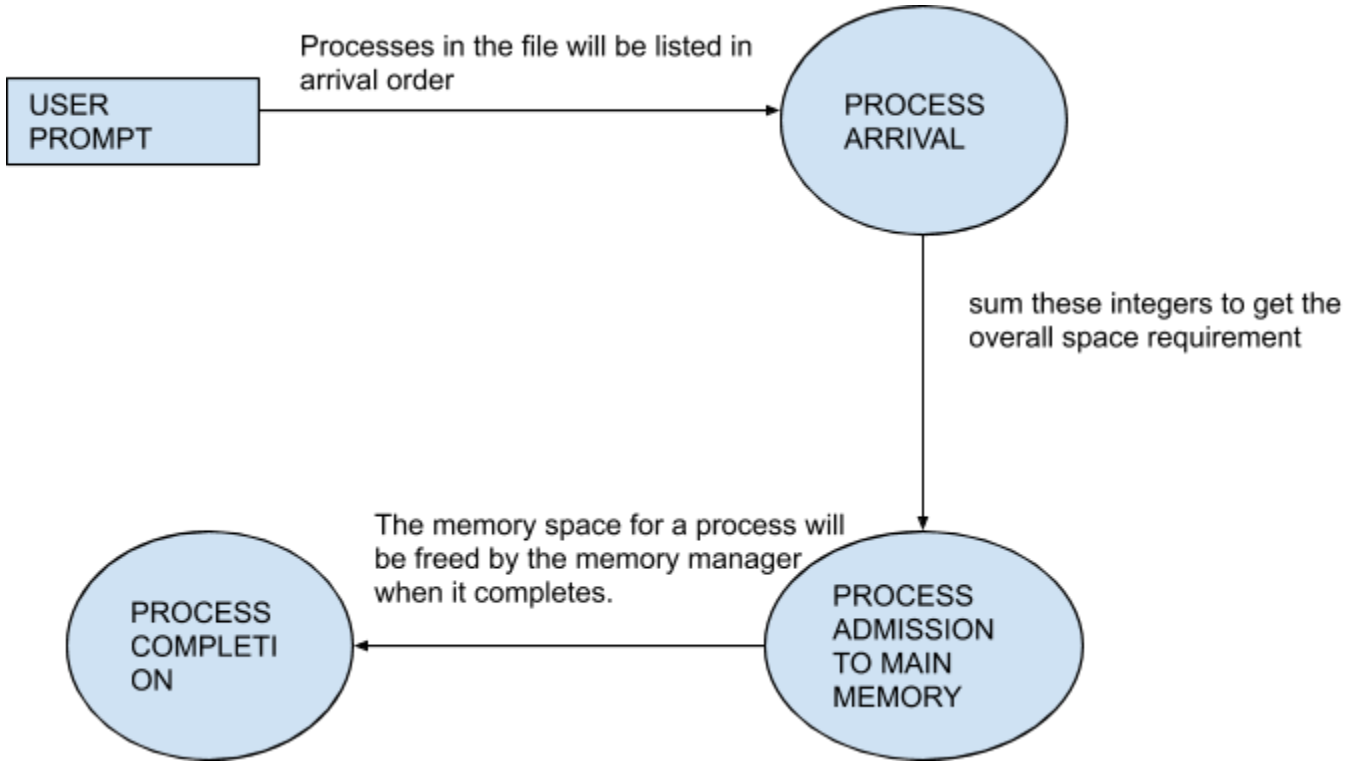
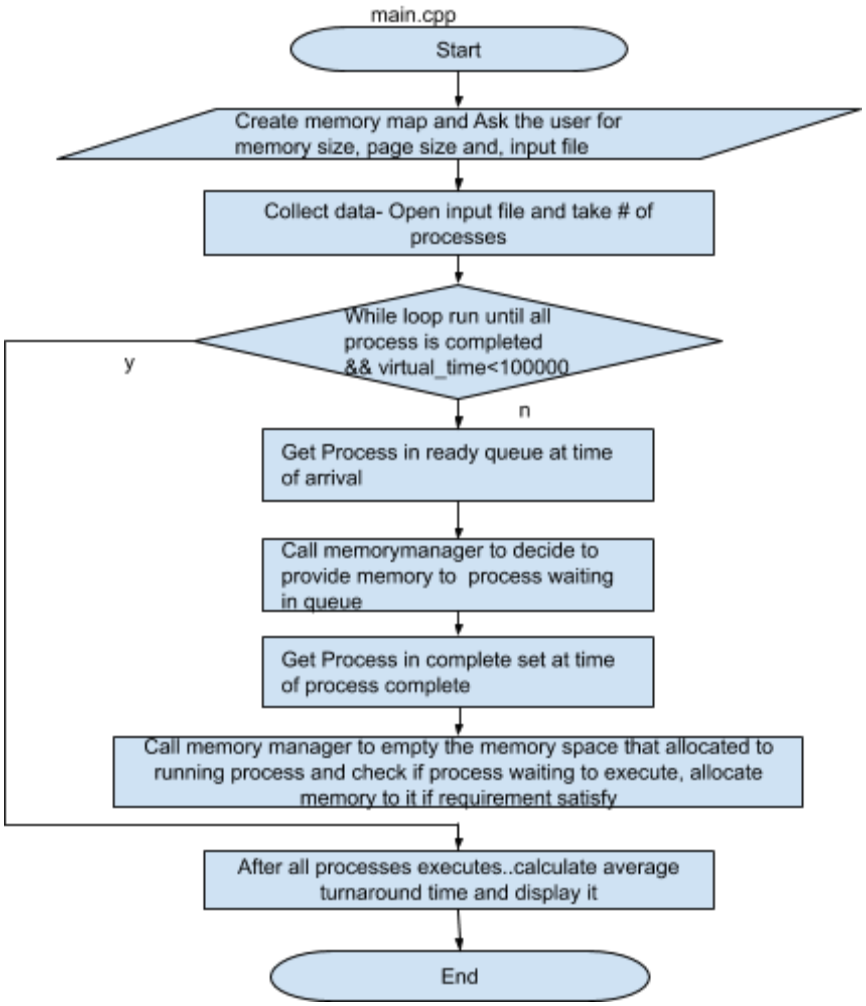
Design of Assignment2
By Janelle Estabillo and Nidhi Shah
Project: Memory Management Simulation

DFD



Flowchart



Pseudocode

main.cpp

Void get_user_input(){

- Ask for memory size
- Ask for page size
- Check if page size and memory size selected is correct

}

Void printmemorymap{

- Print memory map-
- if memorymap.pid is -1.
  Show as free memory.

- Else
  show the frames with page number, process id using the frames

}

void collect_file_data
{

- Open input file
- Get the process id, process arrival time, process termination time(life of process)
- Count the number of memory pieces
- count total memory requirement of the process
- store all information about the process in the vector. And store that vector in process list
- Close file

}

Void displayarrival
{

- Display the time of processes arrived
- Display the list of processes that arrived..(display process Queue)

}

Void memorymanager{

- check the completed time of the process already displayed in the output file or not.. If not ..display it
- Traverse through a vector of the id's of completed processes
- Free the memory previously occupied by the process with id processId
- Remove the process from the In memory process list. (as execution is done)
- print new memory map with removed memory and changed as free memory
- //
- make list of free frames according to page size and memory size
- checking if memorymap.pid is -1 or not.. if is then put those frame list in free frames
- checking queue is empty or not ...if not need those process waiting in queue in memory map
- get value of pageneed according to the page size user choose, and memory requirement of each process in queue
- Allocate frames to process if the memory requirements satisfy, otherwise processes have to wait until get free memory space to use

- Now change those frames from free to used by allocating them to the process
- display the process list that moved to in memory list, as started execution
- Display the new memory map..

}

Int main(){

- create memory map
- get memory_size value and page_size value from user
- create the memory map with -1 as input of array showing is free to use
- get the detail from the file..we know the file so,, not getting filename from user
- now open the output file
- get the processes in the ready queue at time of arrival time
- get the information of memory map and decide to take it out the process from the queue and transfer to currently in memory map
- Get the processes in the completed list at time of completion
- Free the memory from the memory map that is used by the completed process.
- get the information of memory map and decide to take it out the process from the queue and transfer to currently in memory map
- After all processes are executed, count the average turnaround time of this simulation and display it.

}