

Design Document for MP2:

The files that have been modified are:

1. cont_frame_pool.C
2. cont_frame_pool.H

(both the files have descriptive comments for each function)

Here is a detailed overview of the implementations:

1. get_state and set_state functions:
 - a. I have used two bitmaps to describe the state of each frame. One bitmap tells whether the frame is used or free (bitmapUfInfor) while the other tells if the frame is a head or not (bitmapHeadInfo).
 - b. We can use these 2 maps in conjunction to see the frame state and also to set the frame state.
 - c. Get and set has been managed using condition wrt bits in the map.
 - d. This function is expected to set the Hos for stating frame of the requested frames and used for others. Rest frames should remain free status.
2. Constructor (ContFramePool):
 - a. Here we initialize the info frames (the frames where we will store bitmaps in this case)
 - b. We also mark all frames as free
 - c. On the top of b- we mark the frames used to store management info as used
 - d. Now we also initialize the linked list that we are using to store frame pools
 - e. We create a new list for 1st frame pool while we add new node to list if this is not the 1st pool
3. get_frames
 - a. We check if entire pool has required number of free frames
 - b. This function starts with frame 0 and checks if any frame is free.
 - c. Here once we get a free frame we begin another loop to see if the requested number of frames are free
 - d. We run 2nd loop till we get a used frame because in that case we don't have n frames free from that frame index
 - e. We move to next frames and check for free frames that are consecutive
 - f. If we don't get any consecutive free frames we exit
 - g. Also in this case we loop in entire memory so we check if we are exceeding the max limit for available frames at each point to get out of outer loop
 - h. If we get frames then we make 1st of the index as Hos and rest n-1 frames as used frames
 - i. We return the base index of frame set allocated
4. mark_inaccessible
 - a. Here we simple repeat what we did in above. We mark HoS and the used frames if we allocated some frames using get frame
 - b. This can be actually used in get frames but I have just written 2 times
5. release_frames

- The code runs perfectly fine for both kernel and process pools. Here are the screenshots:

