CSCE 611: MP2: Design Document

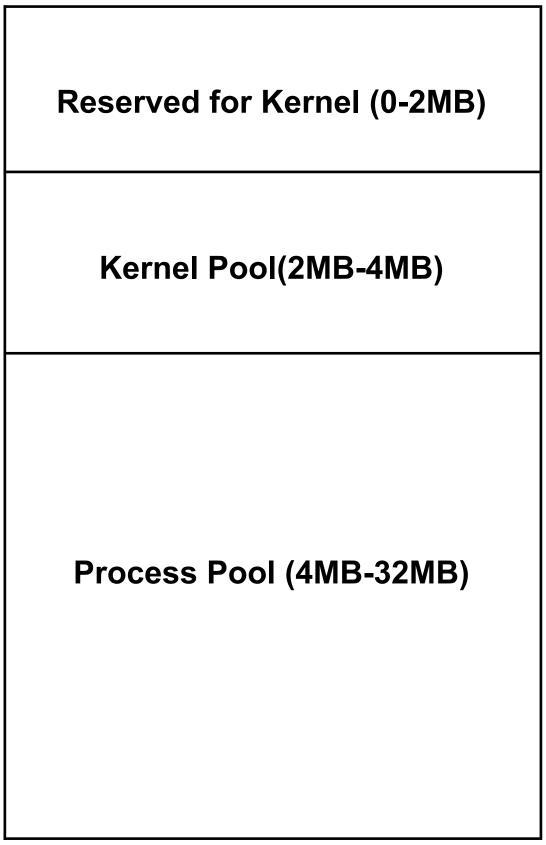
Author: Pranav Anand Taukari

UIN: 433003781

Objective:

The objective of machine problem 2 is to design a frame pool manager which manages allocation and release of frames which are used by kernel frame pool (2MB) and process frame pool (28MB) and mark certain frames as inaccessible (15MB) which are off limits to user.

Schematic representation of memory:



Contiguous Frame Pool Design:

The contiguous frame pool manages kernel frames between 2MB- 4MB and process frames between 4MB- 32MB. It used bit map methodology where two bits of info frames are mapped to a frame which indicates the state of the frame (given below). Also, since two bits are used, one byte of the bitmap can store 4 frames. If the size of the frame is 4KB, then the total number of frames that can be stored in a bitmap is 4KB * 4 = 16K frames.

Bitmap Value	**Description**
11	Represents a free frame
00	Represents a used frame
10	Represents hed of sequence frame

1. **Bitmap implementation:**
   Bitmap is manipulated with two functions namely get_state and set_state with the help of bit operations

   a. **get_state(unsigned long _frame_no):**
      i. Takes a frame number as the input
      ii. Calculates the index of the frame in bitmap and the mask to be used
      iii. Returns the frame state

   b. **set_state(unsigned long _frame_no, Framestate _state):**
      i. Takes the frame number and state as the input
      ii. Calculates the index of the frame in bitmap and the mask to be applied. (Mask for Used and Free is calculated by shifting 0x03 to appropriate position and using XOR and OR operations respectively. Mask for Head of Sequence is calculated by shifting 0x02 to appropriate position and applying AND operation)

iii. Sets the bitmap with appropriate bits (either 11,00,10) based on the frame state

2. **Contiguous Frame Pool Constructor:**
   This constructor initializes the data structure for contiguous frame pool from the starting frame number to base frame number + number of frames. It initializes info_frame_no which is used to indicate the state of the frame using bitmap. It also initializes the linked list which will be utilised to link contiguous frames.

3. **get_frames():**
   This API takes number of frames as arguement and returns the address of the head frame i.e, first frame in the sequence. The function searches for free frames in the pool and reassigns the value to Used. The constraint here is to allocate only contiguous memory. In case non contiguous memory, we begin our search again.
   a. **Detailed Explanation:**
      i. Run a loop upto given number of frames
      ii. Check the state of the frame using get_state()
      iii. If the state of the frame is Free(11) , we decrease the number of frames required
      iv. If the number of frames required becomes 0 we break out of the loop and set the state for head as HoS and other frames as Used in bitmap
      v. Return the address of the head frame

4. **release_frames():**
   The API takes the head frame number as input and returns nothing.It first checks if the given frame number is within the range of the frame pool. If yes, then release frames starting from head frame number until it encounters the next head of sequence.
   a. **Detailed Explanation:**
      i. Identify if the given frame is in the frame pool
      ii. Check if the given frame is head of the sequence
      iii. Mark the head frame as Free
      iv. Mark the rest of frames as Free till you encounter next Head of sequence(HoS)

5. **mark_inaccessible():**
   API takes the base frame number and number of frames as input and returns nothing. This function marks the frames – from base frame number till base frame number + number of frames as head frames to make it as inaccessible.
   a. **Detailed Explanation:**
      i. Same logic as get frames
      ii. Mark the frames as HoS(Head of sequence) to indicate they are inaccessible

6. **needed_info_frames():**
   API takes total number of frames as argument and returns the number of info frames required to store the total frames. In simple pool, 1 bit is used for addressing the frame, so the total number of frames that can be stored in the bitmap is 32k (4K * 8). Similarly, in contiguous pool, we are using 2 bits per frame, so the total frames that can be stored in bitmap are 16k((4k*8)/2). Hence, the total number of info frames is :

   Total number of frames/16*1024 +(Total number of frames%(16*1024)>0 ? 1:0)