Design Document MP2 Operating Systems CSCE 611

Chinmay Ajit Sawkar

Files changed in code:

1. Cont_frame_pool.C - Source file for defining cont_frame pool class with functions like et_frames, mark_inaccessible, release_frames, needed_info_frames
2. Cont_frame_pool.C - header file for Cont_frame_pool.C declaring functions and macros.
3. kernel.C - added code to test the frame pool implementation.

Implementation

Designed a frame manager to get and release frames to be used by the kernel or by user processes. A bitmap is used to track the frame information. I have used 2 bits to determine the frame status (00 if head of frame, 01 if allocated, 00 if free, 00 if inaccessible). A class Cont_frame_pool has been implemented for allocation and management of frames with help of the following functions:

1. constructor:
   Used for getting the arguments and assigning variables like number of first frames managed by this frame pool, size of frame pool, number of the first frame used to store the management information for the frame pool .
   Also used to update the created by the class.

2. get_frames:
   This function searches for a continuous frame in the available frame pool and returns the frame number if an empty continuous sequence is found. For searching I have used linear search but can be optimized to an O(logn) search algorithm. The function also marks the first frame as head of the list and other frames as allocated in bitmap, else if enough continuous frames are not found then returns an appropriate message to the user.

3. mark_inaccessible:
   Used to mark frames inaccessible

4. release_frames :
   Used to free the frames for allocation to different processes. For checking the frame pool to which the frame belongs the function uses the static singly linked list used for tracking the information of the frame pools. The function marks teh bitmap pointer of the frames to be released as free (00).

5. needed_info_frames :
   Used to return the number of frames required to manage a frame pool