Advanced Operating Systems (CSCI-B536) Assignment 5

Hitender Prakash (hprakash@iu.edu)

The assignment 5 is regarding implementing the two more Future flags – SHARED and QUEUE. In FUTURE_SHARED the value produced by one producer is consumed by many consumers. In FUTURE_QUEUE many producer produces the value and many consumers consume that based on their order of execution. Every Producer or Consumer (sharing a common future) produces or consumes only single value.

USAGE: prodcons -f

Files created/Modified:

1. Shell/xsh_prodcons.c: Main Shell code to invoke producer consumer processes
2. Include/future.h: Header file related to Future implementation
3. System/future_alloc.c: code to create Future.
4. System/future _get .c – contains code which is executed by consumer
5. System/future _set.c : Contains code for producer
6. System/f_queue.c : Contains code for handling process queue used by future (Added)
7. Apps/Future_prod.c: Producer entry point.
8. Apps/Future_cons.c: Consumer entry point.

New Functions created/modified:

Modified:
uint future_cons(future*)
future* future_alloc(int)
syscall future_get(future*, int*)
syscall future_set(future*, int*)
uint future_cons(future *fut)

Added:
proc_queue* init_proc_queue();
int f_isempty(proc_queue *head);
f_enqueue (pid32 pid, proc_queue *head);
pid32 f_dequeue(proc_queue *head);

Team:

1. Hitender Prakash (Worked on queue to handle processes and implemented code for FUTURE_QUEUE functionality)
2. Dhaval Niphade (Worked on FUTURE_SHARED handling).