Advanced Operating Systems (CSCI-B536) Assignment 4

Hitender Prakash (hprakash@iu.edu)

The assignment 4 is regarding implementing the Future in Xinu for synchronization and write a producer-consumer program just like we created in the previous assignment for semaphore.

Two processes synchronize using Future. If producer has no value to produce then it will suspend itself after storing its process id in the Future's pid element. When consumer produces new value, it will resume the producer by its pid. Unlike semaphore Future does not have busy waiting.

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 _free.c: Code to free the memory allotted for future
5. System/future _get .c – contains code which is executed by consumer
6. System/future _set.c : Contains code for producer
7. Apps/Future_prod.c: Producer entry point.
8. Apps/Future_cons.c: Consumer entry point.

New Functions created:

uint future_cons(future*)

uint future_prod(future*)

future* future_alloc(int)

syscall future_free(future*)

syscall future_get(future*, int*)

syscall future_set(future*, int*)

Pre-existng Library methods used:

suspend (pid32)

resume(pid32)

freemem(Future*,int)

Team: Assignment is done individually.