6. A Barrier Mechanism in Linux Kernel

key words: kernel thread and synchronization, wait queue, memory barrier, fork, loadable driver module, cross-compilation, system call, patch file.

- Barrier synchronization allows parallel loop computation to start next iteration if all loops have completed the previous iteration.
- In NPTL pthread library, it is realized by 3 functions for threads of the same process, i.e., pthread_barrier_init, pthread_barrier_wait, and pthread_barrier_destroy. (we are not using those, however)
- Develop a barrier synchronization mechanism in Linux kernel which can be invoked by user processes via system call interface.
- Mechanism consists of 3 Linux kernel system calls which are resemble to the pthread barrier:

  1. *barrier_init(unsigned int count, unsigned int *barrier_id, signed int timeout)* – to initiate a barrier in the caller's address space. "count" is the number of threads (i.e., processes and lightweight processes) that are required to be synchronized. If the barrier is initiated successfully, an integer barrier id is returned to the caller process. The *timeout* parameter indicates the maximal waiting time in nanoseconds of any participating threads. A zero or negative value in *timeout* implies there is no timeout in the barrier.

  2. *barrier_wait(unsigned int barrier_id)* – to wait for a barrier synchronization. The caller process will be blocked until the required number of threads have called the wait function, or a timeout occurs, i.e., the barrier synchronization fails.

  3. *barrier_destroy(unsigned int barrier_id)* – to destroyed the barrier of the id *barrier_id.*

- The testing program forks two child processes. In each child process, there are 5 threads to exercise the 1st barrier and additional 20 threads use the 2nd barrier.
- Each thread sleeps a random amount of time before entering a new round of synchronization.
- To make this "new" Linux system calls statically defined and compiled into the kernel, rebuilt the kernel with new system calls.
- Create a patch file that consists of changes to the original kernel source code.