**THREAD MANAGEMENT USING POSIX LIBRARY**

1.Write a program to create 3 threads with the detach and cancel status as below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Thread#** | **Detached (Y/N)** | **Cancel type** | **Cancel status** |
| 1 | Y | **PTHREAD\_CANCEL\_DISABLE** |  |
| 2 | N | **PTHREAD\_CANCEL\_ENABLE** | **PTHREAD\_CANCEL\_DEFERRED** |
| 3 | N | **PTHREAD\_CANCEL\_ENABLE** | **PTHREAD\_CANCEL\_ASYNCHRONOUS** |

1. Let all the threads read and display their detach, cancel type and status and then display thread specific message as below.

T1: Display message in the format as below every 2 secs

<timestamp> Health OK

T2: **Print numbers starting from 1000 in steps of 2 at an interval of 3 secs in format as below.**

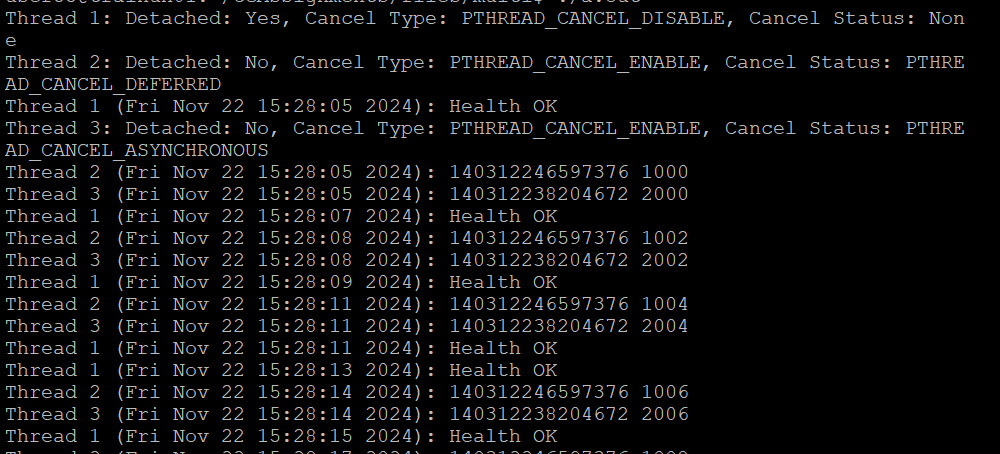
<timestamp> <threadid> <countvalue>

T3: **Print numbers starting from 2000 in steps of 2 at an interval of 3 secs**

<timestamp> <threadid> < countvalue >

1. After creating threads, and after 3 minutes from main(), cancel all 3 threads
2. From an other terminal, use command below to view the thread count of your program
   * + ps -eLF
     + top

[For top command usage to refer <https://www.golinuxcloud.com/check-threads-per-process-count-processes/> ]



1. What difference did you observe between top and ps command?
   * ps -eLF: Lists all threads for processes, showing detailed thread information.
   * top: Displays system processes, including threads, but with less detail than ps -eLF
2. Which column shows the number of threads in ps and in top commands?

* ps -eLF: The NLWP column shows the number of threads.
* top: The THREAD column shows the number of threads in a process.

1. Check the last message timestamp from the threads

* Since the threads are running indefinitely and canceling after 3 minutes, the last timestamp will be based on when the threads are canceled.

1. Which thread was cancelled first and why?

* Thread 1 (detached with cancellation disabled) is the first to exit because it's detached, meaning its resources are automatically freed when it finishes.

1. Were all 3 threads cancelled? Justify the observation

* Yes, all 3 threads were canceled. However, thread 1 might finish its work and exit without being explicitly canceled because it’s detached and doesn’t wait for the main thread to join it.