**Thread Management using POSIX Library - Detachment and Cancellation assignments**

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

a. 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 >

b. After creating threads, and after 3 minutes from main(), cancel all 3 threads

c. 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/ ]

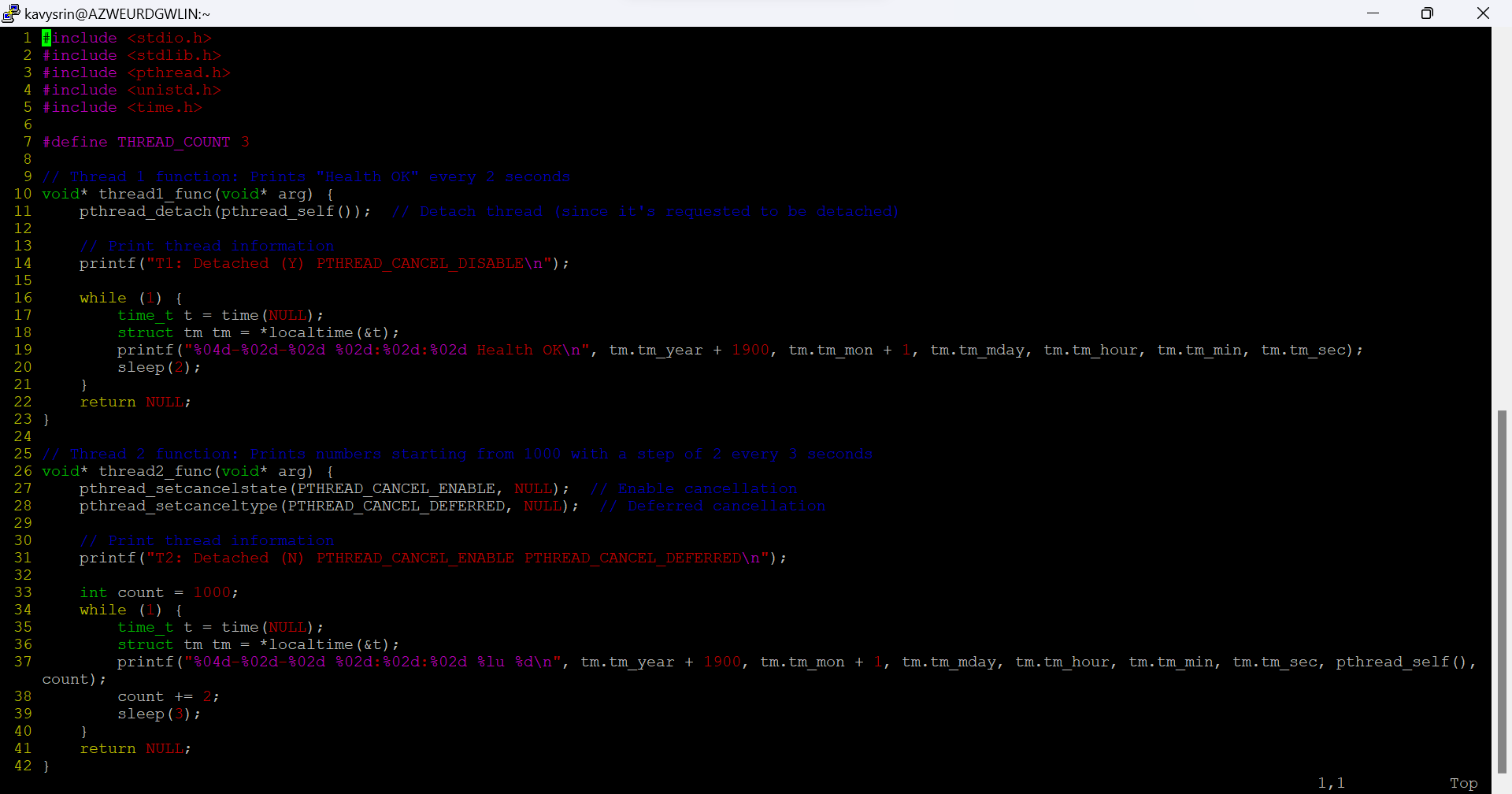
d. What difference did you observe between top and ps command?

e. Which column shows the number of threads in ps and in top commands?

f. Check the last message timestamp from the threads

g. Which thread was cancelled first and why?

h. Were all 3 threads cancelled? Justify the observation



A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

**What difference did you observe between top and ps?**

* ps -eLF shows the number of threads under the NLWP column, whereas top shows threads as separate entries but doesn't always show a consolidated view unless toggled.

**Which column shows the number of threads in ps and top?**

* In ps -eLF, the number of threads is shown in the NLWP column.
* In top, the number of threads is shown as TH under the Tasks section when the thread toggle (H) is enabled.

**Check the last message timestamp from the threads**:

* After cancellation, you will notice that the threads stop printing their respective messages (based on their cancel status and type).

**Which thread was canceled first and why?**

* **Thread 3** with PTHREAD\_CANCEL\_ASYNCHRONOUS is the first to be canceled because asynchronous cancellation takes effect immediately, while deferred cancellation (in Thread 2) requires the thread to reach a cancellation point.

**Were all 3 threads canceled? Justify the observation**:

* **Thread 1** (detached) was not canceled explicitly by the main thread, as it was detached. Detached threads automatically clean up their resources when they finish, but cancellation doesn’t apply to them directly.
* **Thread 2** and **Thread 3** were canceled by the main thread, but due to their cancelation type (deferred vs. asynchronous), their termination behavior was different.