sem_poster.c

```c
/*****************************************************************************
 * Copyright (C) 2023 by Parth Thakkar
 *
 * Redistribution, modification or use of this software in source or binary
 * forms is permitted as long as the files maintain this copyright. Users are
 * permitted to modify this and use it to learn about the field of embedded
 * software. Parth Thakkar and the University of Colorado are not liable for
 * any misuse of this material.
 * ***************************************************************************/

/**
 * @file    sem_poster.c
 * @brief   This program demonstrates posting (incrementing) a POSIX semaphore.
 *          It is typically used in conjunction with another process that waits
 *          on this semaphore. The posting operation signals that an event has
 *          occurred or a resource is available, allowing the waiting process
 *          to proceed.
 *
 *
 * @author  Parth Thakkar
 *
 */

#include <stdio.h>
#include <fcntl.h>    // For O_* constants
#include <sys/stat.h> // For mode constants
#include <semaphore.h>
#include <stdlib.h>

#define SEM_NAME "/semaphore_custom" // Name of the semaphore to post

int main()
{
    // Open the semaphore that has been created by other file
    sem_t *sem = sem_open(SEM_NAME, 0);
    if (sem == SEM_FAILED)
    {
        perror("sem_open failed");
        exit(EXIT_FAILURE);
    }

    // Post the semaphore
    if (sem_post(sem) < 0)
    {
        perror("sem_post failed");
        exit(EXIT_FAILURE);
    }

    printf("Semaphore posted.\n");

    // Close the semaphore to release resources. This operation does not remove
    // the semaphore but detaches it from the process that called sem_close.
    sem_close(sem);

    return EXIT_SUCCESS;
}
```