Question:

Define a global string having two words separated by a space. For example: "abd def". Write two threads such that one of them prints the first word, and the other one prints the second word. You are allowed to define only one function except the main function.

Answer:

Sure, here's a simple implementation in C using POSIX threads:

c code:

#include <stdio.h>

#include <pthread.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_STRING\_LENGTH 100

// Global string

char global\_string[MAX\_STRING\_LENGTH];

// Function to print the first word

void \*print\_first\_word(void \*arg) {

char \*token = strtok(global\_string, " ");

if (token != NULL) {

printf("First Thread: %s\n", token);

} else {

printf("First Thread: No word found\n");

}

pthread\_exit(NULL);

}

// Function to print the second word

void \*print\_second\_word(void \*arg) {

char \*token = strtok(NULL, " ");

if (token != NULL) {

printf("Second Thread: %s\n", token);

} else {

printf("Second Thread: No word found\n");

}

pthread\_exit(NULL);

}

int main() {

// Input the global string

printf("Enter a string with two words separated by a space: ");

fgets(global\_string, MAX\_STRING\_LENGTH, stdin);

// Remove newline character if present

global\_string[strcspn(global\_string, "\n")] = 0;

// Initialize thread IDs

pthread\_t tid1, tid2;

// Create threads

pthread\_create(&tid1, NULL, print\_first\_word, NULL);

pthread\_create(&tid2, NULL, print\_second\_word, NULL);

// Wait for threads to finish

pthread\_join(tid1, NULL);

pthread\_join(tid2, NULL);

return 0;

}

This program defines two threads, each printing one word from the global string. The print\_first\_word function prints the first word, while the print\_second\_word function prints the second word. The main function reads the input string from the user, creates the threads, and waits for them to finish executing before exiting.