

17. Elimination of dead code.

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

pthread_mutex_t mutexA = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t mutexB = PTHREAD_MUTEX_INITIALIZER;

#define TIMEOUT_MICROSECONDS 100000

int try_lock_with_timeout(pthread_mutex_t *mutex) {
    struct timespec timeout;
    timeout.tv_sec = 0;
    timeout.tv_nsec = TIMEOUT_MICROSECONDS * 1000;

    return pthread_mutex_timedlock(mutex, &timeout);
}

void *threadA(void *arg) {
    int retry_count = 0;

    while (retry_count < 5) {
        if (try_lock_with_timeout(&mutexA) == 0) {
            printf("Thread A acquired mutexA\n");

            if (try_lock_with_timeout(&mutexB) == 0) {
                printf("Thread A acquired mutexB\n");
                // Do work with both mutexes
                pthread_mutex_unlock(&mutexB);
            } else {
                printf("Thread A couldn't acquire mutexB\n");
            }

            pthread_mutex_unlock(&mutexA);
            break;
        } else {
            printf("Thread A couldn't acquire mutexA, retrying...\n");
            retry_count++;
        }
    }

    return NULL;
}

void *threadB(void *arg) {
    int retry_count = 0;
```

```

while (retry_count < 5) {
    if (try_lock_with_timeout(&mutexB) == 0) {
        printf("Thread B acquired mutexB\n");

        if (try_lock_with_timeout(&mutexA) == 0) {
            printf("Thread B acquired mutexA\n");
            // Do work with both mutexes
            pthread_mutex_unlock(&mutexA);
        } else {
            printf("Thread B couldn't acquire mutexA\n");
        }

        pthread_mutex_unlock(&mutexB);
        break;
    } else {
        printf("Thread B couldn't acquire mutexB, retrying...\n");
        retry_count++;
    }
}

return NULL;
}

int main() {
    pthread_t tidA, tidB;

    if (pthread_create(&tidA, NULL, threadA, NULL) != 0) {
        perror("pthread_create");
        return 1;
    }

    if (pthread_create(&tidB, NULL, threadB, NULL) != 0) {
        perror("pthread_create");
        return 1;
    }

    pthread_join(tidA, NULL);
    pthread_join(tidB, NULL);

    return 0;
}

```

Output:

C:\Users\hp\OneDrive\Documents\Compiler Design\17.Elimination of dead code..exe

Thread B acquired mutexB

Thread A acquired mutexA

Thread B couldn't acquire mutexA

Thread A acquired mutexB

Process exited after 17.53 seconds with return value 0

Press any key to continue . . .