

BLM 3010.1 Sistem Programlama



MARMARA ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ

BLM 3010.1 PROJE ÖDEVİ 2
BAKLAVA HAZIRLAMA YAZILIMI

Hazırlayanlar

170419512 Mehmet Kaan Erol

1. Giriş

Bu projede bakalava hazırlama uygulaması üzerinden çok thread'li bir programın senkronizasyonunun sağlanması hedeflenmiştir. Uygulamada 6 usta ve 1 toptancı thread'ler ile temsil edilmektedir. Toptancı ana thread'i ustalar ise oluşturulan thread'lere karşılık gelir. Amaç semaphore ile thread'ler arasında senkronizasyonu sağlamaktır.

2. Uygulama

Uygulama Ubuntu 20.04 işletim sistemi üzerinde C99 standardında geliştirildi. Geliştirme sırasında derleyici olarak GCC 9.4.0 versiyonu kullanıldı. Uygulamada kullanılan thread'ler ve semaphore'lar POSIX API'sine aittir.

2.1 Kod İçeriği

Bu başlık altında programın sahip olduğu fonksiyonlar hakkında bilgi verilmektedir.

2.1.1 main() Fonksiyonu

Program bu fonksiyondan çalışmaya başlar, komut satırından verilen argümanlar bu fonksiyona parametre olarak geçirilir. Argümanlar `getopt()` hazır fonksiyonu ile parçalanır. Verilen argümandan dosya adı alınır ve bir char dizisinde saklanır. Argümanların alınması, dosyanın içeriğini tutulması, signal'in özelleştirilmesi, thread'lerin oluşturulması, ustalara ait ihtiyaçların belirlenmesi burada yapılır. Program `toptanci()` fonksiyonunun bitmesiyle beraber yine main'de kaldığı yere döner, ustalar ait thread'lere özelleştirilmiş signal'i gönderir ve sonlanır.

2.1.2 usage() Fonksiyonu

Bu fonksiyonda programın kullanımı hakkında bilgilendirme amacıyla bir `stderr` tanımlayıcısı ile çıktı verilir.

2.1.3 readFile() Fonksiyonu

Parametre olarak dosya adı alınır, dosyanın var olup olmadığı kontrol edilir. Dosya var ise 'r' ile okuma modunda açılır ve okuma işlemi başlar. Okuma sırasında her bir satır tek seferde alınması için geçici bir 3 byte'lık buffer oluşturuldu. Her seferinde bu buffer kadar yer okunup önceden tanımlanan kaynak struct'ına atandı. Burada zorluk dosyada kaç satır içeriğin olduğunun bilinmemesiydi. Bu, dinamik belleğin boyutunun dinamik olarak artırılarak çözülebilir ancak bizim uygulamamız için gereksiz olacaktı. Bu nedenle önce dosya bir kere açıldı, satır sayısı alındı ardından tekrar okuma yapılarak atama işleme tamamlandı. Elde edilen dinamik dizinin referansı return ile döndürüldü.

2.1.4 usta() Fonksiyonu

Bu fonksiyon ustayı temsilen oluşturulan thread'lere aittir. Uygulamada ustalar için yapılması istenen tüm işlemlere sahiptir. Tüm thread'ler için ortaktır, ustalar fonksiyona argüman olarak geçirilen usta bilgisi ile ayrıştırılıyor. Bu argümanda hem ustayı temsil eden sayı hem de ustanın ihtiyaçları tutulmaktadır. Burada semaphore ile senkronizasyonun sağlanması hedeflenmiştir. Dört farklı isimde semaphore tanımlandı, bunlar: toptancının kaynağı, malzeme teslim, malzema ataması, baklava hazır durumlarını tutmaktadır.

2.1.5 toptanci() Fonksiyonu

Uygulamada istenen toptanci rolünün işlevlerine sahip fonksiyondur. Ana thread'de çalışır. Amacı dosyadan okunan kaynakların usta thread'lere dağıtmaktır. Bu esnada tutarsızlık oluşmaması için semaphore ile thread'ler arasında senkronizasyonun sağlanması hedeflenmiştir. Kaynaklar bittikten sonra fonksiyon tamamlanır, main fonksiyona dönülür ve usta thread'lere signal gönderilerek tuttıkları kaynakları serbest bırakmaları ve çıkmaları sağlanır.

3. Değerlendirme

Uygulama hazırlanan kaynak dosya ile test edildi, valgrind ve tsan (thread sanitizer) araçları ile değerlendirildi. Proje dizininde bulunan kaynak dosya ile yapılan testlerde proje dökümanında istendiği şekilde senkronizasyonun sağlandığı görüldü. Programın herhangi bir bellek sızıntısına ya da segmentasyon hatasına sahip olup olmadığını değerlendirmek amacıyla valgrind aracı kullanıldı. Bu araç ile elde edilen çıktılar Şekil 1'de gösterilmiştir. Çıktılarda herhangi bir bellek sorunu ile karşılaşılması.

```
kaanubuntu@DESKTOP-FKCTEH2:/mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2$ valgrind --leak-check=full
./program -i kaynaklar.txt
==3342== Memcheck, a memory error detector
==3342== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==3342== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==3342== Command: ./program -i kaynaklar.txt
==3342==
Usta 5, S ve F malzemeleri için bekliyor.
Usta 4, S ve U malzemeleri için bekliyor.
Usta 3, F ve G malzemeleri için bekliyor.
Usta 6, U ve F malzemeleri için bekliyor.
Usta 1, S ve G malzemeleri için bekliyor.
toptanci U ve F teslim etmektedir.
Usta 2, U ve G malzemeleri için bekliyor.
Usta 6, U ve F malzemelerini aldı.
Usta 6 baklavayı hazırlıyor.
toptanci baklavayı bekliyor.
Usta 6 baklavayı toptanciya teslim etti.
Usta 6, U ve F malzemeleri için bekliyor.
toptanci baklavayı aldı ve satmaya gitti.
toptanci S ve U teslim etmektedir.
Usta 1, S ve G malzemelerini aldı.
toptanci baklavayı bekliyor.
Usta 1 baklavayı hazırlıyor.
Usta 1 baklavayı toptanciya teslim etti.
Usta 1, S ve G malzemeleri için bekliyor.
toptanci baklavayı aldı ve satmaya gitti.
Kaynaklar bitti. Program kapanıyor...
==3342==
==3342== HEAP SUMMARY:
==3342==   in use at exit: 1,632 bytes in 6 blocks
==3342==   total heap usage: 12 allocs, 6 frees, 7,331 bytes allocated
==3342==
==3342== 1,632 bytes in 6 blocks are possibly lost in loss record 1 of 1
==3342==   at 0x483DD99: calloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==3342==   by 0x40149CA: allocate_dtv (dl-tls.c:286)
==3342==   by 0x40149CA: _dl_allocate_tls (dl-tls.c:532)
==3342==   by 0x485C322: allocate_stack (allocatestack.c:622)
==3342==   by 0x485C322: pthread_create@@GLIBC_2.2.5 (pthread_create.c:660)
==3342==   by 0x1099FA: main (in /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/program)
==3342==
==3342== LEAK SUMMARY:
==3342==   definitely lost: 0 bytes in 0 blocks
==3342==   indirectly lost: 0 bytes in 0 blocks
==3342==   possibly lost: 1,632 bytes in 6 blocks
==3342==   still reachable: 0 bytes in 0 blocks
==3342==   suppressed: 0 bytes in 0 blocks
==3342==
==3342== For lists of detected and suppressed errors, rerun with: -s
==3342== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
kaanubuntu@DESKTOP-FKCTEH2:/mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2$
```

Şekil 1. valgrind ekran görüntüsü

Tsan aracı ile de race durumu oluşup oluşmadığı kontrol edildi. Kod tsan ile ilk derlendiğinde Şekil 2'de gösterildiği gibi data race hatası ile karşılaşıldı.

```

kaanubunt@DESKTOP-FKCTEH2:/mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2$ gcc -g 170419512_baklava.c -o program -pthread -fsanitize=thread
kaanubunt@DESKTOP-FKCTEH2:/mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2$ ./program -i kaynaklar.txt
Usta 1, S ve G malzemeleri için bekliyor.
Usta 2, U ve G malzemeleri için bekliyor.
Usta 3, F ve G malzemeleri için bekliyor.
Usta 4, S ve U malzemeleri için bekliyor.
Usta 5, S ve F malzemeleri için bekliyor.
Usta 6, U ve F malzemeleri için bekliyor.
toptanci U ve F teslim etmektedir.
=====
WARNING: ThreadSanitizer: data race (pid=3758)
  Write of size 4 at 0x5636f86c4030 by main thread:
    #0 toptanci /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:69 (program+0x16a6)
    #1 main /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:238 (program+0x1ffa)

  Previous read of size 4 at 0x5636f86c4030 by thread T1:
    #0 usta /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:101 (program+0x17b0)

  Location is global 'isSourceReady' of size 4 at 0x5636f86c4030 (program+0x000000005030)

  Thread T1 (tid=3760, running) created by main thread at:
    #0 pthread_create ../../src/libsanitizer/tsan/tsan_interceptors_posix.cpp:962 (libtsan.so.0+0x5ea79)
    #1 main /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:230 (program+0x1fa2)

SUMMARY: ThreadSanitizer: data race /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:69 in toptanci
=====
WARNING: ThreadSanitizer: data race (pid=3758)
  Read of size 1 at 0x7b0400000004 by thread T6:
    #0 usta /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:102 (program+0x17f3)

  Previous write of size 1 at 0x7b0400000004 by main thread:
    #0 toptanci /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:61 (program+0x15c4)
    #1 main /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:238 (program+0x1ffa)

  As if synchronized via sleep:
    #0 usleep ../../src/libsanitizer/tsan/tsan_interceptors_posix.cpp:355 (libtsan.so.0+0x62f3b)    #1 usta /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_baklava.c:104 (program+0x17a4)

  Location is heap block of size 8 at 0x7b0400000000 allocated by main thread:
    #0 malloc ../../src/libsanitizer/tsan/tsan_interceptors_posix.cpp:651 (libtsan.so.0+0x38323)    #1 main /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_baklava.c:225 (program+0x1f3e)

  Thread T6 (tid=3765, running) created by main thread at:
    #0 pthread_create ../../src/libsanitizer/tsan/tsan_interceptors_posix.cpp:962 (libtsan.so.0+0x5ea79)
    #1 main /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:230 (program+0x1fa2)

SUMMARY: ThreadSanitizer: data race /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_170419512_proje2/170419512_baklava.c:102 in usta
=====

```

Şekil 2. Tsan ekran görüntüsü data race sorunu

Sorunun nedeni Şekil 3'te gösterilmiştir. Sorun kaynakların doğruluğunu karşılaştırmak için erişilen readySource referansına aynı zamanda *toptanci()* fonksiyonunda erişiliyordu. Bu durum data race'e neden oluyordu. Sorun yine Şekil 3'te gösterildiği gibi binary semaphore ile çözüldü.

Sorunu görüldüğü yer:

```

while (isSourceReady == 1 && ((neededSources->s1 == readySource->s1 && neededSources->s2 == readySource->s2) ||
    (neededSources->s1 == readySource->s2 && neededSources->s2 == readySource->s1)))
    usleep(50);

if (status)

```

Sorun Çözümü:

```

// malzemeler hazır mı?
sem_wait(&semAssignSource);
status = ((neededSources->s1 == readySource->s1 && neededSources->s2 == readySource->s2) ||
    (neededSources->s1 == readySource->s2 && neededSources->s2 == readySource->s1));
usleep(50);
sem_post(&semAssignSource);

```

Şekil 3. Data race sorunun kaynağı ve çözümü

Sorunun kaynağı bulunduktan sonra tekrar tsan ile kod derlendi ve çıktısı beklendiği gibi sorunsuzdu. Tsan çıktısı Şekil 4'te gösterilmiştir.

```
kaanubuntu@DESKTOP-FWCTEH2: /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_178419512_proje2$ gcc -g 178419512_baklava.c -o program -pthread -fsanitize=thread
kaanubuntu@DESKTOP-FWCTEH2: /mnt/c/Users/kaane/Documents/Courses/bm3_2/systemProgramming/mehmetkaanerol_178419512_proje2$ ./program -i kaynaklar.txt
toptanci U ve F teslim etmektedir.
Usta 6, U ve F malzemelerini aldı.
Usta 6 baklavayı hazırlıyor.
toptanci baklavayı bekliyor.
Usta 6 baklavayı toptanciya teslim etti.
toptanci baklavayı aldı ve satmaya gitti.
Usta 6, U ve F malzemelerini aldı.
Usta 6 baklavayı hazırlıyor.
toptanci S ve U teslim etmektedir.
toptanci baklavayı bekliyor.
Usta 4, S ve U malzemelerini aldı.
Usta 4 baklavayı hazırlıyor.
Usta 6 baklavayı toptanciya teslim etti.
toptanci baklavayı aldı ve satmaya gitti.
toptanci S ve F teslim etmektedir.
```

Şekil 4. Sorunsuz tsan ekran görüntüsü