Зэрэгцээ байдал ба Хяналтын хийсвэрлэл

Даалгавар 1: (1 оноо)

Энэ даалгаварт Java зэрэгцээ программын хэсгүүдийг агуулсан таван файл өгсөн. Эхний мөрүүдэд байгаа зарлалт нь программ эхлэх үед хувьсагчдыг дараалсан байдлаар үүсгэдэг гэж үзнэ. Үргэлжлээд "thread1", "thread2"методууд тус тусдаа урсгал (thread) болж биелэгдэх ба программ нь тэдний дуусахыг "join()"методоор хүлээнэ.

Файл бүр кодын төгсгөлд дөрвөн өгсөн асуулттай. Эхний асуулт нь өгөгдлийн уралдаан (race condition) байгаа эсэхийг асуух бөгөөд "ТИЙМ"эсвэл "ҮГҮЙ"гэж хариулна. Дараа өгөгдлийн уралдаан гарч буй мөрийн дугаарыг бичих ёстой. Мөн уралдааны нөхцөлд орж буй хувьсагчийн нэрсийг тодорхойлох шаардлагатай. Эцэст нь, биелэлтийн төгсгөлд тухайн хувьсагчийн бүх боломжит утгыг тодорхойлж бичнэ. Example.txt файлыг хариултын жишээ болгон ашиглаарай.

Даалгавар 2: (2 оноо)

Энэ даалгаварт *үйлдвэрлэгч-хэрэглэгч (producer-consumer)*-ийн системийн код өгөгдсөн бөгөөд кодын зөв ажиллагааг хангахын тулд mutex ашигласан синхрончлолыг нэвтрүүлэх зорилготой.

Үйлдвэрлэгч-хэрэглэгчийн систем нь гурван үйлдвэрлэгч, гурван хэрэглэгчээс бүрдэнэ. P1, P2, P3 гэж тэмдэглэгдсэн үйлдвэрлэгчид R, G, B тэмдэгтүүдийг үүсгэж, тэдгээрийг `producer_array` руу нэмдэг. Энэ массив нь бүх урсгалуудад хэрэглэгдэх дундын хувьсагч байна. P1 нь R ба G, P2 нь G ба B, P3 нь B болон R-ийг үүсгэдэг. Үйлдвэрлэгчид producer_array-ийн төгсгөлд үүсгэж буй тэмдэгтүүдээ нэмэхдээ RGBRGB...RGB гэсэн дараалаар оруулна.

C1, C2, C3 гэж тэмдэглэгдсэн хэрэглэгчид `producer_array`-аас сүүлийн тэмдэгтийг устгаад, уг тэмдэгтээ `consumer_array`-руу нэмдэг. Энэ массив нь бүх хэрэглэгчийн урсгалуудад хэрэглэгдэх дундын хувьсагч. `consumer_array`-ын дараалал нь RGBRGB...RGB хэлбэрийг хангаж байгаа эсэхийг хянах ёстой.

Энэ системийн кодыг task_2 хавтас доторх main.c файлд өгсөн. Дараах командыг ашиглан кодыг компиляц хийнэ: `gcc -pthread main.c -o main`. Компиляц хийгдсэн файлыг ажиллуулсны дараа `producer_array` болон `consumer_array`-ын аль алинд нь үүсгэсэн дараалал төлөвлөсөн хэлбэртэй нийцэхгүй байгааг ажиглах болно (RGBRGB...).

Таны даалгавар бол код нь төлөвлөсөн гаралтыг бий болгохын тулд mutexes ашигласан синхрончлолыг нэвтрүүлэх юм. Системийн логикийг өөрчлөх эсвэл одоо байгаа кодын мөрөнд өөрчлөлт оруулахгүй гэдгийг анхаарна уу. **Зөвхөн синхрончлолын кодын мөрүүд, өөрөөр хэлбэл, mutex зарлалт болон түүнийг хэрэглэх хэсэг л нэмэх боломжтой**.

Санамж: pthread сангаас pthread_mutex_t-г ашиглана. Программын код C17, GCC 9.4 дээр ажиллах боломжтой байх ёстой.

Даалгавар 3: (2 оноо)

Энэ даалгавар нь одоо байгаа дараалсан C програмуудыг OpenMP ашиглан параллел болгоно. Дөрвөн жижиг C програм өгсөн бөгөөд зорилго бол тэдгээрийг параллел болгохын тулд тохирох OpenMP pragma-уудыг оруулах юм. Параллел ажиллах ёстой программын хэсгийг програмын эх кодод комментоор зааж өгсөн болно. task_3 хавтас доторх файлуудыг өөрчилнө.

Анхны болон парллел болгосон програмуудыг хоёуланг нь компиляц хийхийн тулд дараах командыг ашиглана: `gcc -fopenmp my_file.c -o my_prog`, энд `my_file.c` нь компиляц хийх эх кодын файлын нэрийг, харин `my_prog` нь компиляц хийгээд үүсэх файлын нэрийг илэрхийлнэ.

Санамж: Параллел программуудыг C17 болон GCC 9.4 хувилбарыг ашиглан компиляц хийнэ. Программын семантикийг параллел болгохоос өөрөөр өөрчлөхгүй байх, OpenMP pragma-аас өөр код нэмэх/хасахгүй байх ёстой.

Даалгавар 4: (2 оноо)

Энэ даалгавар нь long-running функцийг удирдахдаа promises болон async/await ашиглахад чиглэгддэг. task_4 хавтсанд callbacks болон/эсвэл promises ашигласан функцийн хэрэгжилтийг агуулсан файлууд байгаа. Зорилго бол эдгээр функцийг дараах байдлаар өөрчлөх юм.

- program_1: Callback-ын оронд promises-ыг ашиглахын тулд `ExecuteTasks` доторх кодыг өөрчилнө. Файлын төгсгөлд тестийн жишээ өгөгдсөн.
- program_2: Callback-ын оронд async/await-г promise-оор ашиглахын тулд `executeTasks` доторх кодыг өөрчилнө. Файлын төгсгөлд тестийн хоёр тохиолдлыг өгсөн болно.
- program_3: `HTTPRequest, parseResponse, processData` функцууд нь promises буцаадаг. Даалгавар бол `fetchData` функцийн кодыг callback хэлбэрээс promise хэлбэр болгон хувиргах юм. Файлын төгсгөлд гарах ёстой үр дүн, тестийн хоёр тохиолдлыг өгсөн.
- program_4: Promises ашиглан бичсэн функц өгөгдсөн. Зорилго бол `getUserInfo`, `getUserComments`, `getUserPosts` функцуудыг `await`-тэй дуудах кодыг хөгжүүлнэ. Өөрөөр хэлбэл, `await` механизм ашиглан `getUserDataOld` функцийн өөр хэрэгжилтийн код бичнэ. Файлын төгсгөлд тестийн хоёр тохиолдлыг өгсөн болно.

Санамж: Илүү дэлгэрэнгүй мэдээллийг task_4 хавтас дахь файл тус бүрд өгсөн коммент, зааврааас авна уу.

ТАНД АМЖИЛТ ХҮСЬЕ!