Ei-reentranttien kirjastokutsujen ongelma liittyy tilanteisiin, joissa useat osaohjelmat tai säikeet voivat kutsua samaa kirjastofunktiota samanaikaisesti, ja tämä kirjastofunktio ei ole suunniteltu käsittelemään samanaikaisia kutsuja turvallisesti. Reentrantti kirjastofunktio sen sijaan on suunniteltu niin, että se voi käsitellä useita samanaikaisia kutsuja turvallisesti.

Ongelma voi ilmetä, kun ei-reentrantti kirjastofunktio käyttää jaettuja muuttujia tai resursseja, jotka voivat tulla ristiriitaan, kun useat kutsijat yrittävät käyttää niitä samaan aikaan. Tämä voi johtaa odottamattomiin ja vaikeasti jäljitettäviin virheisiin ohjelmassa.

Esimerkki:

Oletetaan, että meillä on yksinkertainen ei-reentrantti kirjastofunktio, joka laskee lukuja peräkkäin ja palauttaa aina seuraavan luvun:

int nextNumber = 0;

int getNextNumber() {

int result = nextNumber;

nextNumber++;

return result;

}

Jos useat osaohjelmat tai säikeet kutsuvat tätä getNextNumber-funktiota samanaikaisesti, ne voivat aiheuttaa ongelmia. Esimerkiksi, jos säie A kutsuu getNextNumber-funktiota ja saa tulokseksi 5, mutta ennen kuin se palauttaa tuloksen, säie B kutsuu myös getNextNumber-funktiota ja saa saman tuloksen 5, sekaannus voi syntyä. Säikeet A ja B voivat molemmat käyttää väärin tuloksen 5 ja se voi johtaa virheisiin ohjelman toiminnassa.

Reentrantti versio tällaisesta kirjastofunktiosta olisi suunniteltu niin, että jokainen säie tai osaohjelma saa oman erillisen kopion nextNumber-muuttujasta tai käytetään jollain muulla tavalla synkronointia varmistaakseen, että samanaikaisten kutsujen aiheuttamat ristiriidat vältetään.

Tämä esimerkki kuvastaa ei-reentrantin kirjastokutsun ongelmaa, kun useat toimijat yrittävät käyttää jaettuja resursseja ilman asianmukaista synkronointia.