

Korszerű Programozási Technikák

KisZH IV

A feladatban fájlok letöltését és a letöltött adatok ellenőrzését szimuláljuk. A `_data` adattag a letöltött adatokat jelzi, nem kell foglalkoznod vele. A `download` metódus a letöltendő fájl nevét kapja paraméterként és szimulálja a fájl letöltését (a szál időközönkénti elaltatásával). Az `integrityCheck` függvény pedig a letöltött adatok integritását ellenőrzi és visszatér a rossz adatblokkok számával. A feladat szálak és kölcsönös kizárás alkalmazásával megvalósítani egy fájl letöltését és ellenőrzését. Mindegyik esetben figyelj arra, hogy a szálak a megfelelő helyeken vissza legyenek csatolva a fő szálhoz.

1. A `main` függvényben a `download` metódust indítsd el egy külön szálon, paraméterként a `file_name` változót átadva. (1 pont)
2. Készíts egy másik függvényt `progress_check` néven a `main.cpp`-be, ami 500 milliszekundumonként ellenőrzi, hogy a letöltés fut-e még. Amíg fut írja ki, hogy a letöltés még folyamatban van, a letöltés végén pedig írja ki, hogy végzett a példaként megadott kimenetnek megfelelően. A függvényt indítsd el egy külön szálon a letöltés megkezdése után, paraméterként a fájl nevét is add át (a futás ellenőrzéséhez használj közös `mutex` objektumot, és megfelelően módosítsd a `download` függvényt is). (2 pont)
3. A letöltés befejezése után írd meg egy `integrity_check_lambda` nevű lambda függvényt, ami a `capture-list`-ben megkapja a `bad_count` változót, paramétere nincs, visszatérési típusa `void`, a függvénytorzsban pedig a `bad_count` értékét beállítja az `integrityCheck` függvény segítségével. A lambda függvényt futtasd egy külön szálon a `main` függvényben. (2 pont)