Lefedettségmérés a gcc –finstrument-functions kapcsolója segítségével

Mi is ez a -finstrument-functions

- Gcc egy kapcsolója
- Lehetővé teszi a kód profilozását
  - Függvények futásidő mérése
  - függvényhívások gyakorisága
- Beavatkozási pontokat hozhatunk létre
  - Függvények előtt és után
  - Saját logikával bővíthetjük

```c
// Instrumentation functions
void __attribute__((no_instrument_function)) __cyg_profile_func_enter(void *, void *func_addr) {
    clock_gettime(CLOCK_MONOTONIC, &start_time); // Az időzítő elindítása a függvénybelépési ponton
    function_call_count++;
}

void __attribute__((no_instrument_function)) __cyg_profile_func_exit(void *, void *func_addr) {
    clock_gettime(CLOCK_MONOTONIC, &end_time); // Az időzítő leállítása a függvénykilépési ponton
    double elapsed_time = (end_time.tv_sec - start_time.tv_sec) + (end_time.tv_nsec - start_time.
    tv_nsec) / 1e9; // Az eltelt idő kiszámítása másodpercben
    printf("Function ran for %.6f seconds\n", elapsed_time);
}

// Example functions
void foo() {
    printf("Inside foo()\n");
    // Egy kis várakozás hozzáadása a futási idő méréséhez
    for (int i = 0; i < 100000000; i++) {
        // Üres ciklus csak a várakozáshoz
    }
}

void bar() {
    printf("Inside bar()\n");
}

int main() {
    foo();
    bar();

    printf("Total function calls: %d\n", function_call_count);
```

```
C:\Users\kispal\Desktop\tesztelés>gcc -finstrument-functions test_runtime.c -o test_runtime

C:\Users\kispal\Desktop\tesztelés>.\test_runtime.exe
Inside foo()
Function ran for 0.244817 seconds
Inside bar()
Function ran for 0.001480 seconds
Total function calls: 3
Function ran for 0.003077 seconds

C:\Users\kispal\Desktop\tesztelés>
```

Lefedettségmérés

- egy olyan technika, mellyel az mérhető, hogy a kód mely részei futottak le

```
C:\Users\kispal\Desktop\tesztelés>test_demo
Entered function at address 00007FF6C90A1340
main
Entered function at address 00007FF6C90A1A08
test_add
Entered function at address 00007FF6C90A18FC
function add
Exited function at address 00007FF6C90A18FC ran for 0.001186 seconds
Test add passed!
Exited function at address 00007FF6C90A1A08 ran for 0.002257 seconds
Entered function at address 00007FF6C90A1A0D
test_subtract
Entered function at address 00007FF6C90A1983
function subtract
Exited function at address 00007FF6C90A1983 ran for 0.001049 seconds
Test subtract passed!
Exited function at address 00007FF6C90A1A0D ran for 0.002254 seconds
Exited function at address 00007FF6C90A1340 ran for 0.004516 seconds
```

Amit tudni kell

- kapcsoló használata lehetővé teszi részletes futásidői követést és adatok gyűjtését, amelyek segíthetnek az alkalmazás elemzésében és optimalizálásában
- A „gcov” képes arra, hogy pontosabb és részletesebb adatokat szolgáltasson
- hook függvényeknek a használata jelentős hatással lehet az alkalmazás futásidőre és teljesítményére

Előnyök	Hátrányok
Futásidői követés	**Teljesítménycsökkenés**
Követési adatok gyűjtése	**Nagyobb bináris méret**
	Megbízhatósági problémák

Köszönöm a figyelmet