

# **Výkonnostní testy knihovny FString**

**Projekt do předmětu IPA**

# 1. Testovací prostředí

Testy proběhly na zařízení s procesorem *Intel® Core™ i5-4690 CPU @ 3.50GHz* × 4 na 64 bitové verzi *Ubuntu 18.04.1 LTS*.

Pro jazyk C byl použit překladač *GCC 7.3.0* s vypnutou optimalizací (*-O0*) a přepínačem *-no-pie*.

Pro jazyk Python byl použit interpret *Python 3.6.3*.

## 2. Obsah testu

Obsah testu a realizace v jazyce C bez použití fstring, v jazyce Python a v jazyce C s použitím fstring:

### 1. Načtení statického řetězce do dynamického řetězce

- C: *malloc, strlen, strcpy*
- Python: *=*
- C FString: *fstrfromstr*

### 2. Převod všech písmen na malá

- C: *for, tolower*
- Python: *lower*
- C FString: *fstr\_to\_lower*

### 3. 4000x připojení začátečního textu na konec řetězce

- C: *for, realloc, strlen, strcpy*
- Python: *for, range, +=*
- C FString: *for, fstrappend*

### 4. Převod všech písmen na velká

- C: *for, toupper*
- Python: *upper*
- C FString: *fstr\_to\_upper*

### 5. Vložení textu (s posunutím zbytku) na index 1

- C: *realloc, for, strlen, =*
- Python: *=, [:], +*
- C FString: *fstrinsert*

### 6. Vložení textu na konec řetězce

- C: *realloc, strlen, strcpy*
- Python: *+=*
- C FString: *fstrappend*

### 7. Nalezení vloženého textu

- C: *strstr*
- Python: *find*
- C FString: *fstr\_find\_first*

### 8. Vypsání nalezeného indexu pro kontrolu správnosti

- C: *printf*
- Python: *print, format*
- C FString: *prinf*

### 9. Uvolnění paměti

- C: *free*
- Python:
- C FString: *fstrfree*

### 3. Výsledky testu

Doba běhu celého programu byla spočítána utilitou *time*.

Všechny programy vygenerovaly stejný výstup, ale počet funkcí a řádků se lišil:

Jazyk C bez knihovny FString: 25 volání funkcí

Jazyk Python: 7 volání funkcí/metod (nejsou započítány magické metody)

Jazyk C s použitím knihovny FString: 10 volání funkcí

#### 3.1 Program napsaný v jazyce C bez použití knihovny FString

Doba běhu: 435 ms

```
real    0m0.435s
user    0m0.430s
sys     0m0.004s
```

#### 3.2 Program napsaný v jazyce Python

Doba běhu: 22 ms

Rychlost: **19.8x rychlejší** než jazyk C bez FString

```
real    0m0.022s
user    0m0.018s
sys     0m0.004s
```

#### 3.3 Program napsaný v jazyce C s použitím knihovny FString

Doba běhu: 3 ms

Rychlost: **7.3x rychlejší** než jazyk Python

**145x rychlejší** než jazyk C bez FString

```
real    0m0.003s
user    0m0.000s
sys     0m0.002s
```

### 4. Závěr

Testy prokázaly, že vektorizovaný přístup a ručně optimalizovaný assembler kód má velký vliv na rychlost programu a samotný testovací kód prokázal, že správně napsané rozhraní knihovny může snížit počet použitých funkcí a tedy řádků kódu a zvýšit přehlednost kódu.