**Звіт-Ретроспектива**

на лабораторну роботу №3

Волчецького Руслана

SMAZ

Smaz - це проста бібліотека, придатна для стиснення дуже коротких рядків (string, char\*).

За допомогою цієї бібліотеки я планував розробити повноцінну програму для стиснення текстів, статтей, і інших видів текстових даних де (загалом) переважають букви алфавіту (будь-якого) - і це в мене вийшло.

Чому ж я обрав саме цю бібліотеку? Відповідь очевидна. Бібліотеки стиснення загального призначення створюють необхідний стан для динамічного стиснення даних, щоб мати змогу стискати всі види даних. Це дуже гарна ідея, але не для конкретної проблеми: стиснення маленьких рядків не будуть працювати. Натомість SMAZ стискає текст в середньому на 40-50% і навіть може стиснути рядок розміром 3 байти!

(Для прикладу “the” він стискає у 1 байт).

Початок роботи з бібліотекою був надзвичайно легкий - ніяких проблем з її пошуком, встановленням та налаштуванням не виникло. Так само зрозумілою була і документація цієї бібліотеки - вона не містить нічого зайвого і одночасно корисна.

З використанням бібліотеки також проблем не було, тому що вона не містить надмірної кількості функцій, методів і тд, з якими прийшлось би розбиратись тривалий час. Тим не менш ніякого надлишкового коду писати не прийшлось - я відразу приступив до розробки ідеї самої програми.

Поведінка функцій бібліотеки була досить передбачувана, хоча звичайно прийшлось почитати інструкції що саме вона робить, що приймає та що повертає (без інструкцій нікуди, дякую розробнику).

Взаємодія між функціями бібліотеки не є складною для розуміння - все ж тут дає свій відбиток їх мала кількість, що є дуже зручним. А ось при написанні власного коду приходилось декілька разів перевіряти чи дійсно якась конкретна функція виконає те ж, що і від неї очікується . Але це не зайняло багато часу, все ж таки всі дії були передбачувані.

При використанні бібліотеки звичайно не обійшлось без проблем.

Першою і головною проблемою стало використання char\* та string і їх вбудованих методів. Діло в тому що при виконанні compressing-а рядка функції автора використовують символи ‘\0’ та ‘\n’ (останній врешті-решт не виявився проблемою). Чому це погано? Діло в тому що після compressing-а в вихідному рядку можуть міститись декілька символів ‘\0’ які аж ніяк не позначають кінець рядка. Тому функція strlen практично працювала некоректно - адже вона видавала лише довжину до першого такого символу. На щастя функція при виконанні повертає довжину отриманого compressed рядка, тому деякими “маніпуляціями” все ж таки я зміг в нормальному вигляді використовувати string та char\* (В загальному випадку він працює для статичного char[]).

Другою проблемою стало зчитування compressed тексту з файлу. Діло в тому, що (як потім виявилось) функції також використовують і символ EOF в функції compressing. Це було жахливо розуміти - адже тепер символ EOF міг знаходитись прямо всередині compressed тексту! Тому прийшлось трохи попрацювати щоб повноцінно і правильно зчитувати дані з такого файлу (Зокрема я використовую розмір файлу в байтах, щоб дізнатись скільки саме мені потрібно зчитати з файлу).

Хорошого в цій бібліотеці багато чого - зокрема працює вона так як і обіцялось - тексти стискаються на 45% в середньому, і без проблем розтискаються в безперечно той самий текст.

Серед поганого виділю використання ‘\0’ та EOF при стисканні, що робить роботу з текстом набагато важчою.

В підсумок можу сказати : якби мені прийшлось розробляти аналогічну бібліотеку, я б не використовував деякі символи при цьому - так, це б зробило стискання дещо менш ефективним (на декілька відсотків), але полегшило б роботу з самим стиснутим текстом в декілька разів. Тому затрати часу тут невиправдані.

Посилання на ресурси:

<https://github.com/antirez/smaz> - офіційна документація

<https://cpp.libhunt.com/smaz-alternatives> - порівняння з іншими видами

<https://stackoverrun.com/ru/q/660823> - пошук таких бібліотек