

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №3

Програмування взаємодії з ОС

Прийняв:

Викладач

Ковтунець О.В.

«___»_____2023 р.

Виконав:

Студент 3 курсу, гр. ІК-03

Смірнов Андрій

Олександрович

2023 р.

Мета роботи

Ознайомлення з довкіллям розробки в Unix-подібних ОС і системними викликами в таких ОС, отримання навичок написання програм для Unix-подібних ОС, вивчення методу взаємодії з ОС за допомогою системних викликів.

Загальні відомості до виконання роботи

Результатом виконання роботи є програма для ОС Linux, написана мовою програмування C, яка опрацьовує надані текстові файли (логи)

- ☐ файл 01-tiny-access.log;
- ☐ файл 05-huge-access.log;

у форматі **ACCESS LOG** веб-сервера **APACHE** і демонструє відповідну вашому варіантові інформацію. Ця програма повинна використовувати ті ж самі утиліти, які використовувались в л.р. №2 (cat, grep, sort, head, awk, uniq, cut, paste тощо), але логіку роботи оболонки (shell) потрібно реалізувати за допомогою

операторів мови C і системних викликів (використати виклики fork, варіантивиклику exec, wait або waitpid, open, close, pipe, dup2).

Під логікою роботи оболонки (shell) мається на увазі:

- ☐ запуск процесів і очікування їх результатів,
- ☐ відкриття файлів, перенаправлення вводу-виводу, pipe,
- ☐ умовні вирази і цикли.

Також в C можна реалізувати підрахунок агрегованих значень і форматний вивід.

Хід роботи

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_DATE_LENGTH 11

typedef struct {
    char date[MAX_DATE_LENGTH];
    int count;
} LogEntry;

void updateCount(LogEntry* entries, int size, const char* date) {
    int i;
    for (i = 0; i < size; i++) {
```

```

        if (strcmp(entries[i].date, date) == 0) {
            entries[i].count++;
            return;
        }
    }

    // Дата не знайдена, додати новий запис
    strcpy(entries[size].date, date);
    entries[size].count = 1;
}

int compareEntries(const void* a, const void* b) {
    int countA = ((LogEntry*)a)->count;
    int countB = ((LogEntry*)b)->count;

    return countB - countA;
}

void printTopDates(LogEntry* entries, int size) {
    int i;
    int totalCount = 0;

    // Загальна кількість звернень
    for (i = 0; i < size; i++) {
        totalCount += entries[i].count;
    }

    printf("Top 10 Dates:\n");
    for (i = 0; i < size && i < 10; i++) {
        double percentage = (entries[i].count / (double)totalCount) *
100;
        printf("%d. %s - %d - %.2lf%%\n", i + 1, entries[i].date,
entries[i].count, percentage);
    }
}

char* extractDate(const char* line) {
    const char* dateStart = strstr(line, "[") + 1; // Знаходимо
позицію початку дати
    char* date = (char*)malloc(MAX_DATE_LENGTH * sizeof(char));
    strncpy(date, dateStart, MAX_DATE_LENGTH - 1);
    date[MAX_DATE_LENGTH - 1] = '\0'; // Завершуємо рядок нуль-
терміном

```

```

        return date;
    }

int main() {
    FILE* file = fopen("01-tiny-access.log", "r");
    if (file == NULL) {
        perror("Unable to open file");
        return 1;
    }

    LogEntry entries[100]; // Припустимо, що максимальна кількість
    різних дат - 100
    int entryCount = 0;

    char line[256];
    while (fgets(line, sizeof(line), file)) {
        // Отримання дати з рядка
        char* date = extractDate(line);

        updateCount(entries, entryCount, date);

        free(date); // Звільняємо пам'ять, виділену для дати
    }

    fclose(file);

    // Сортювання в порядку спадання
    qsort(entries, entryCount, sizeof(LogEntry), compareEntries);

    // Виведення топ-10 дат
    printTopDates(entries, entryCount);

    return 0;
}

```

Цей код використовує функцію **extractDate**, яка виділить з рядка лог-файлу. Припускається, що дата завжди починається з першого символу рядка і має фіксовану довжину 10 символів. Функція **extractDate** виділяє пам'ять для нового рядка, копіює дату та повертає вказівник на виділену пам'ять. Після використання дати її слід звільнити, використовуючи **free**.

```
1. 09/Oct/2016 - 358 - 10,47%
2. 10/Oct/2016 - 348 - 10,18%
3. 01/Oct/2016 - 347 - 10,15%
4. 20/Oct/2016 - 346 - 10,12%
5. 25/Oct/2016 - 339 - 9,91%
6. 24/Oct/2016 - 337 - 9,85%
7. 12/Oct/2016 - 337 - 9,85%
8. 19/Oct/2016 - 336 - 9,82%
9. 06/Oct/2016 - 336 - 9,82%
10. 02/Oct/2016 - 336 - 9,82%
```

Вивід для 01-tiny-access.log

```
1. 29/Nov/2016 - 35046 - 10,56%
2. 23/Nov/2016 - 34390 - 10,37%
3. 24/Oct/2016 - 33089 - 9,97%
4. 08/Nov/2016 - 32995 - 9,95%
5. 15/Nov/2016 - 32912 - 9,92%
6. 22/Nov/2016 - 32911 - 9,92%
7. 07/Nov/2016 - 32878 - 9,91%
8. 14/Nov/2016 - 32811 - 9,89%
9. 21/Nov/2016 - 32400 - 9,77%
10. 03/Nov/2016 - 32295 - 9,74%
```

Вивід для 05-huge-access.log