

ПП_Т_04. Паралельні обчислення

ПП_Т_04.1 Описати клас-нащадок Thread, який забезпечує передачу необроблених виключень, що виникли у потоці, потоку, який запустив даний. Для цього передбачити окремий метод. Перевірити виконання для потоку обчислення факторіалу, коли аргумент не є цілим числом.

ПП_Т_04.2 Описати програму з графічним інтерфейсом, у якому користувач вводить номер n числа Фібоначчі, а програма викликає рекурсивну функцію обчислення n -го числа Фібоначчі. Функція має виконуватись у окремому потоці. Після завершення обчислень результат має бути показаний також у графічному інтерфейсі.

Графічний інтерфейс має породжувати нове вікно для кожного введеного числа Фібоначчі і в цьому вікні показувати результат після завершення обчислень.

ПП_Т_04.3 Один потік через випадкові інтервали від 1 до t_1 надає повідомлення. Інший потік має обробити ці повідомлення. Обробка займає від 1 до t_2 . Скласти програму, яка генерує та обробляє повідомлення. Повідомлення мають оброблятися у порядку їх надходження. Обробка повідомлення – це просто показ його на екрані. Використати чергу.

ПП_Т_04.4 В умовах попередньої задачі вважати, що є пріоритетні повідомлення, які мають бути оброблені раніше, ніж звичайні. Забезпечити таку обробку. Описати власний клас на базі стандартного класу deque, який забезпечує розміщення та подальшу обробку повідомлень.

ПП_Т_04.5 В умовах попередньої задачі для обробки повідомлень описати власний клас, у якому використати 2 черги (об'єкти стандартного класу Queue).

ПП_Т_04.6 Описати програму з графічним інтерфейсом, у якому користувач вводить номер ціле число, а програма викликає функцію що перевіряє, чи є це число простим. Функція має виконуватись у окремому потоці. Після завершення обчислень результат має бути показаний також у графічному інтерфейсі.

Графічний інтерфейс має породжувати нове вікно для кожного введеного простого числа і в цьому вікні показувати результат після завершення обчислень.

ПП_Т_04.7 У супермаркеті є N кас. У деякий момент часу працюють m касирів. У черзі до касирів стоять k покупців. Новий покупець додається до черги до деякої каси через випадковий інтервал від 1 до t_1 . Касир обслуговує покупця за випадковий інтервал від 1 до t_2 . Через випадковий інтервал від 1 до t_3 один з касирів йде на перерву на час від 1 до t_4 . Якщо черга зростає,

адміністрація додає резервного касира, до якого переходять покупці з інших черг. Зафіксувати параметри N , m , k , t_1 , t_2 , t_3 , t_4 . Для цих параметрів шляхом моделювання розрахувати приблизно потрібну кількість резервних касирів та довжину черги, при якій слід додавати резервного касира, щоб максимальна довжина черги до 1 касира не перевищила l_{\max} .

Вважати, що загальна кількість кас N є достатньою, щоб задовольнити умову. Також вважати, що покупці самостійно рівномірно розподіляються між чергами.

Вказівка: одна черга до каси – це 1 потік.

ПП_Т_04.8 Є готель з N номерами. Через випадковий час від 1 до t_1 приходить новий клієнт та заселяється у один з номерів (якщо є вільний), у якому живе час від 1 до t_2 . Якщо вільних номерів немає, то клієнт очікує на звільнення будь-якого номера. При заданих параметрах N , t_1 , t_2 розрахувати середню довжину черги клієнтів та середній час очікування.

Вказівка: один клієнт – це 1 потік.

ПП_Т_04.9 Є стадіон на N глядачів з m турнікетами. Турнікети відчиняють за t до початку матчу. Через 1 турнікет глядач проходить за випадковий час від 1 до t_1 . Усі глядачі приходять до стадіону до початку матчу у випадкові моменти часу але з рівномірним розподілом. Змоделювати проходження глядачів через турнікети та розрахувати, за який мінімальний час до початку матчу треба прийти, щоб з імовірністю 0.9 пройти на стадіон до початку матчу при заданих значеннях параметрів.

Вказівка: один турнікет – це 1 потік.

ПП_Т_04.10 Скласти програму, яка має 2 потоки. Основний потік у процесі роботи має створювати у деякому каталозі тимчасові файли з розширенням .tmp. Другий потік-демон через певний проміжок часу t_1 перевіряє цей каталог та видаляє файли з розширенням .tmp, які були змінені не пізніше ніж за час t_2 до моменту перевірки.

ПП_Т_04.11 Залізниця має одноколіїну ділянку. Цю ділянку поїзди проходять за деякий випадковий час у діапазоні від t_1 до t_2 . Якщо у момент підходу зустрічного потяга одноколіїна ділянка зайнята, то зустрічний потяг чекає її звільнення. Промоделювати рух потягів, вважаючи що наступний потяг з кожного напрямку підходить до ділянки через випадковий інтервал від t_3 до t_4 . Кожний потяг – це 1 потік.

Використати черги.

ПП_Т_04.12 В умовах задачі ПП_Т_04.11 використати окремі потоки та пул потоків.

ПП_Т_04.13 В умовах задачі ПП_Т_04.11 використати блокування.

ПП_Т_04.14 Скласти програму, у якій є основний процес та підпроцес. Підпроцес генерує текстові файли великого розміру та додає їх до деякого каталогу Каталог1. У основному процесі головний потік аналізує цей каталог і якщо з'явився новий файл або декілька нових файлів, то ініціює перевірку, чи містить цей файл рядок, що відповідає деякому шаблону. Перевірку здійснюють додаткові потоки. Додатковий потік після перевірки переміщує файли, що містять вказаний шаблон, у Каталог2, а ті, що не містять, - у Каталог3. Використати пул з n додаткових потоків.

ПП_Т_04.15 В умовах задачі _____ змінити процес роботи наступним чином. Допоміжні потоки працюють постійно. Основний потік, коли з'явився новий файл, додає завдання до черги, яку розбирають додаткові потоки. Використати окремі потоки та чергу.

ПП_Т_04.16 Один процес через випадкові інтервали від 1 до t_1 надає повідомлення. Інший процес має обробити ці повідомлення. Обробка займає від 1 до t_2 . Скласти програму, яка генерує та обробляє повідомлення. Повідомлення мають оброблятися у порядку їх надходження. Обробка повідомлення – це просто показ його на екрані. Використати чергу.

ПП_Т_04.17 Описати програму з графічним інтерфейсом, у якому користувач вводить номер n числа Фібоначчі, а програма викликає рекурсивну функцію обчислення n -го числа Фібоначчі. Функція має виконуватись у окремому процесі. Після завершення обчислень результат має бути показаний також у графічному інтерфейсі.

Графічний інтерфейс має породжувати нове вікно для кожного введеного числа Фібоначчі і в цьому вікні показувати результат після завершення обчислень.

ПП_Т_04.18 Описати програму з графічним інтерфейсом, у якому користувач вводить номер ціле число, а програма викликає функцію що перевіряє, чи є це число простим. Функція має виконуватись у окремому процесі. Після завершення обчислень результат має бути показаний також у графічному інтерфейсі.

Графічний інтерфейс має породжувати нове вікно для кожного введеного простого числа і в цьому вікні показувати результат після завершення обчислень.

ПП_Т_04.19 Скласти програму, яка має 2 процеси. Основний процес у своїй роботі має створювати у деякому каталозі тимчасові файли з розширенням `.tmp`. Другий процес через певний проміжок часу t_1 перевіряє цей каталог та видаляє файли з розширенням `.tmp`, які були змінені не пізніше ніж за час t_2 до моменту перевірки.

ПП_Т_04.20 Скласти програму, яка перевіряє правильність навчальних програм студентів. Кожна студентська програма написана у Python, має певну кількість вхідних даних та певний 1 результат. Вхідні дані вводяться з клавіатури. Результат – це останній рядок, що виводиться на екран. Результатами можуть бути цілі, дійсні числа або рядки (звичайно, що на екран усе виводиться у вигляді рядка).

Створювана програма має запустити студентську програму у окремому процесі, передати їй вхідні дані, почекати визначений максимальний час завершення, проаналізувати отриманий результат. Має бути проаналізовано: чи завершено програму у визначений максимальний час, чи були необроблені виключення під час виконання програми, чи правильно отримано результат (для результатів дійсного типу проаналізувати близькість до очікуваного значення).

Параметри для перевірки містяться у окремому текстовому файлі – у кожному рядку – по 1 завданню. Параметри мають вигляд:

<шлях_до_програми> (<аргументи>) (<тип даних>:<результат>)

Наприклад:

c:\mydir\gcd.py (5,10) (int:5)