



СПОСІБ ГЕНЕРАЦІЇ БІНАРНИХ ДАНИХ ЗА ДОПОМОГОЮ СКРИПТОВИХ МОВ ПРОГРАМУВАННЯ ДЛЯ ДЕЯКИХ ЗАВДАНЬ У ГАЛУЗІ ПОЛІГРАФІЇ

Сокорчук І.П., старший викладач, катедра ІІ, ХНУРЕ

У сучасній поліграфічній промисловості бінарні дані є невід'ємною частиною процесу створення та друку друкованих матеріалів. Вони використовуються для зберігання різноманітної інформації, зокрема, такої як зображення, шрифти, макети, дані кольорів та управління друкарським обладнанням. Але при розв'язанні складних завдань тут можуть виникнути труднощі в управлінні та обробці таких даних, особливо через різноманітність програмних засобів та форматів, що може призвести до проблем з сумісністю та інтероперабельністю між різними поліграфічними системами.

Скриптові мови програмування відомі своїм широким застосуванням у сучасних обчислювальних системах. У поліграфії вони можуть використовуватися для автоматизації багатьох рутинних операцій, таких як підготовка файлів для друку, обробка зображень, розмітка тексту. Скриптові мови використовуються для автоматизації процесу керування кольорами у друкарських системах, допомагають автоматизувати процес розмітки тексту та керування шрифтами у документах. Скриптові мови дозволяють автоматизувати процеси обробки та підготовки зображень для друку, включаючи зміну роздільної здатності, кадрування, коригування кольорів тощо. Ці мови використовуються для автоматизації процесів друку та керування друкарським обладнанням, таким як налаштування друкарських параметрів, керування друкарськими чергами тощо. Вони мають ряд переваг, мають підтримку передових технік програмування, наявність розвинутого вбудованого інструментарію, здатність до вирішення задач за допомогою меншого обсягу програмного коду, а також мінімальну залежність від програмно-апаратної платформи. Проте треба відзначити, що скриптові мови характеризуються переважно слабкою типізацією або використанням динамічної типізації даних, що ускладнює роботу з даними, які вимагають жорсткої статичної типізації, зокрема з бінарними даними. У таких випадках, можна використовувати особливості окремих скриптових мов або використовувати призначені для роботи з бінарними даними програмні компоненти. Однак цей підхід може призвести до втрати деяких переваг скриптових мов.

Автор запропонував спосіб, який дозволяє ефективно працювати із бінарними даними у скриптах і водночас зберегти переваги скриптових мов програмування. Цей спосіб полягає у тому, що скрипт генерує не самі бінарні дані, а представлений у текстовому вигляді опис цих бінарних даних. Далі цей текст передаються через канали міжпроцесорного обміну в окремий спеціальний програмний постпроцесорний модуль, який перетворює цей текстовий опис бінарних даних у самі бінарні дані та передає їх на обробку в наступні програмні модулі. Для цього автор трозробив мову для опису бінарних даних, а також функції



та команди постпроцесора, призначені для керування обробкою та генерацією даних. Для опису даних використовується стандартний формат, зазвичай прийнятий у багатьох мовах програмування. Числові дані можуть представлятися у різних системах числення, таких як десяткова, вісімкова, шістнадцяткова тощо. Текстові дані описуються як окремі символи або рядки символів (табл. 1).

Таблиця 1 – Команди опису даних

Двійкові	b10110110 b0111011101001111
Вісімкові	03 0235 01234567
Десяткові	2345678 37 65536
Шістнадцяткові	0xFA1E 0xABCD 0x123
Символьні	'A' '\n' '\t'
Рядкові	"Data" "TEST"

Команди управління обробкою даних дозволяють налаштовувати розмірність бінарних даних, встановлювати порядок байтів у багатобайтових числових даних, а також конвертувати текстові дані у вихідний формат (табл. 2).

Таблиця 2 – Команди управління даними

Розрядність	%8 %16 %32 %64
Послідовність байтів	%BIG %LITTLE
Кодова таблиця	%UTF8 %CP1251 %IBM866
Розмір блоків	%BLOCK:1024:B1

Постпроцесор також включає в себе лічильники байтів, обчислювачі контрольних сум для блоків даних, а також функції для додавання додаткових бінарних даних до вихідних даних за допомогою встановлених шаблонів. Це дозволяє розширити функціональність системи, заповнювати або вирівнювати блоки даних до потрібного розміру та додавати різні контрольні суми. Управління параметрами лічильників, обчислювачів та генераторів даних здійснюється за допомогою спеціально розроблених команд управління генерацією даних (табл. 3).

Таблиця 3 – Команди генерації даних

Обчислення контрольних сум	%CRC8:S1 %CRC16:S2 %CRC32:S3
Лічильники байтів	%COUNT:C4
Заповнення блоків	%FILL:B1
Генерація даних	%DATA:S1

Описані рішення автор використав у промисловій комп'ютерній системі для налаштування устаткування та управління ним.

Список літератури

1. Сокорчук І.П. (2017). Спосіб роботи із бінарними даними у скриптових мовах програмування на Unix-сумісних платформах. *Радіоелектроніка та молодь у XXI столітті.* (с. 90-91).
2. Сокорчук І.П. (2007). Комп'ютерна програма «Вимірювально-обчислювальний комплекс автоматизованої системи обліку енергоресурсів Promenergy/E7». Свідectво про реєстрацію авторського права на твір № 21713 від 15.08.2007 р.