УДК 004.2:004.5

Крак Ю.В., д.ф.-м.н., проф.,
Коваль Ю.В., асистент,
Ставровський А.Б., к.ф.-м.н., доцент

Віртуальний процес: означення та застосування в створенні системи жестового інтерфейсу

Київський національний університет імені Тараса Шевченка, 83000, м. Київ, пр-т. Глушкова 4д,
e-mail: kafedraTK@unicyb.kiev.ua

Iu. V. Krak, Prof,
Iu. V. Koval, Assistant,
A. B. Stavrovskyi, Assoc.prof.

Virtual process: definition and application for gestures interface system creation

Taras Shevchenko National University of Kyiv, 83000, Kyiv, Glushkova ave., 4d,

e-mail: kafedraTK@unicyb.kiev.ua

В статті розглянуто поняття процесу та віртуального процесу. Запропоновано узагальнення поняття віртуальний процес та основи нотації віртуальних процесів. Розглянуто адресний простір як середовище виконання процесу, простір імен як відповідне загальне середовище віртуального процесу та зазначено відмінності простору адрес та простору імен. В якості прикладу віртуального процесу розглянуто загальну архітектуру системи для інтерактивного жестового інтерфейсу комунікації людина-комп'ютер та описано відповідний віртуальний процес. Запропоновано паралельне незалежне виконання на різних платформах частин головної системи та їх асинхронне керування.

Ключові слова: віртуальний процес, простір імен, жестова мова, інтерфейс жестової мови.

Process and virtual process term were discussed. Proposed an extension for virtual process term. The syntax and semantics of some virtual process operators were proposed, the main architecture construction was described as virtual process using proposed operators. Other virtual process operators will be discussed in the future. Address space and name space were discussed and compared. The main features of both kinds of space were mentioned and advantages and disadvantages of name space were declared. A general architecture of sign language human-computer interface system was discussed and was represented as an example of virtual process. Sign language interface system architecture was divided into three levels: data adaptation level, functional modules level, and design making, synchronisation and control level. Parallel independent execution on different computing platforms of functional modules in the main system and their asynchronous control were proposed.

Key Words: virtual process, name space, sign language, sign language interface.

Статтю представив д.ф.-м.н., проф. Анісімов А.В.

1. Вступ

Процесом у програмуванні зазвичай називають власне процес виконання програми. За принципами фон Неймана (von Neumann) побудови комп'ютера, процес розміщується в оперативній пам'яті комп'ютера й у ньому є код (команди, що виконуються процесором) та значення (оброблювані дані) [1]. Пам'ять комп'ютера має адресацію, і відповідно до цього процес фактично розміщується в адресному просторі. Команди та значення мають адреси, за допомогою яких їх можна використати. Команди та значення розрізняються процесором тільки під час виконання коду. Коли процес завершується, пам'ять вивільняється й адресний простір зникає. Проте для інтерпретованих програм наявність адресного простору не є обов'язковою. В таких процесах ідентифікація змінних може відбуватися за іменами й, відповідно, змінні розміщуються не в адресному просторі, а в просторі імен.

Альтернативою описаному процесу є поняття віртуального процесу [2,3], що базується на однопоточній моделі, з можливістю паралельного виконання. Компонент управління планує виконання

інших компонентів і передає їм управління. Планування може виконуватися періодично або на основі подій. Компоненти, виконання яких заплановано, відповідають за виконання запланованих дій і повертають управління в компонент управління. Ця модель заснована на логічному розкладанні дій на прості кроки, для виконання яких потрібні тільки короткі проміжки часу. Програміст несе відповідальність за розкладання дій компонентів відповідно до вимог до програми. Прикладом такого процесу є утиліта make, яка, як правило, керує побудовою коду, викликаючи компілятор, компонувальник та інші програми.

2. Розширення поняття віртуальний процес

У віртуальному процесі, описаному вище, його складові частини можуть бути лише однопоточними й обмінюватися даними тільки з компонентом управління. Розширимо це поняття в такий спосіб. Віртуальний процес – це сукупність кодів та значень, що можуть виконуватися та оброблятися на різних, навіть архітектурно, обчислювальних системах з метою розв'язання певної задачі за певним алгоритмом. Нижче це поняття буде уточнено, а спочатку розглянемо два приклади.

1. На веб сервері для повноцінної роботи використовується база даних, яка сама є сервером. Ця алгоритмічно регламентована взаємодія серверів є прикладом віртуального процесу. Більше того, якщо керування обома серверами відбувається за допомогою браузера (і це на сьогодні є типовим), то робота браузера та серверів також складає віртуальний процесс, наприклад бухгалтерського обліку.

2. У клієнт-серверній архітектурі браузер – це програма першого користувача, веб сервер – другого. Перший дає другому запит на отримання даних, які браузер потім, можливо, відображує. Користувачів першого та другого типів може бути скільки завгодно, а вибір даних, що передаються, є хаотичним і, як наслідок, неалгоритмізованим. Тут є взаємодія різних процесів, а не віртуальний процес, оскільки акти взаємодії веб сервера та клієнта не алгоритмізуються.

Складовими частинами віртуального процесу мають бути звичайні процеси. Але й віртуальні процеси можуть бути складовими елементами більшого віртуального процесу.

Уточнимо поняття віртуального процесу.

1. Віртуальний процес спрямований на алгоритмізовану обробку значень і зберігає значення й, можливо, частини коду в просторі імен.
2. Звичайний процес є атомарним віртуальним процесом.
3. Процеси роботи апаратних пристроїв, що генерують потоки значень або керуються складовими віртуального процесу, є атомарними віртуальними процесами.
4. Спільна обробка значень у двох або більше віртуальних процесах утворює віртуальний процес.
5. В межах віртуального процесу один складовий віртуальний процес може запустити інший складовий віртуальний процес.
6. Коди складових віртуальних процесів можуть генеруватися в межах віртуального процесу.
7. Складові віртуальні процеси можуть тимчасово припиняти свою роботу, проте, стартувавши знову, вони мають доступ до спільного простору імен, де зберігаються значення, і продовжують загальний процес їх обробки.
8. Деякі складові віртуальні процеси можуть аварійно завершуватися, міняти свій код, і це не призводить до припинення віртуального процесу.
9. Віртуальний процес зникає, якщо руйнується його простір імен.

Віртуальний процес має команди й значення, проте не має адресного простору. Отже, віртуалізація процесу не моделює об'єкт, як, наприклад, віртуальна пам'ять, а лише відтворює його головну функцію – виконання команд для обробки значень.

Атомарний процес віртуального процесу може мати доволі складну реалізацію, навіть з застосуванням віртуалізації, проте для головного віртуального процесу, якщо немає

жодних втручань у внутрішню структуру, він залишається атомарним.

Простір імен суттєво відрізняється від простору адрес. Причина в тому, що нумерація адрес відбувається за допомогою натуральних чисел і, як наслідок, існують сусідні елементи. Для простру імен, де нумерація відбувається лексикографічними одиницями, відсутнє поняття сусідства елементів, незважаючи на упорядкованість. Отже, втрачається можливість переходу від елемента до сусіднього йому, проте з'являється ряд беззаперечних переваг.

1. Можливість вставити елемент між двома сусідніми.
2. Неможливість, за доступності деякого елемента, мати доступ до сусіднього елемента, що підвищує надійність зберігання значень.
3. Можливість утворення незалежних підпросторів імен.

3. Постановка задачі

Як зазначено в [4], для побудови систем інтерфейсу на базі жестової інформації (зокрема, розпізнавання елементів жестової мови) можуть бути застосовані різні підходи й методи. Кожен із цих методів має переваги й недоліки, тому не існує алгоритму, що є найкращим за всіх можливих вхідних даних. Звідси, в основу роботи системи, яка отримує інформацію шляхом обробки потоку даних, кладеться конкурентне розпізнавання за кількома різними алгоритмами. Отже, маємо таку постановку задачі: розробити метод та здійснити алгоритмічну реалізацію віртуального процесу розпізнавання елементів жестової мови.

4. Метод побудови віртуальних процесів

Віртуальний процес розв'язання задачі в якості атомарних процесів має реалізації окремих алгоритмів розпізнавання елементів жестової мови. Його схему, запропоновану в [4], наведено на рис. 1. Система складається з трьох частин:

– розпаралелювач вхідного потоку, що готує дані для подальшої обробки та аналізу;

– M_1, …, M_n – функціональні та логічні модулі;

– аналізатор подій – система прийняття рішень, керування та синхронізації модулів.

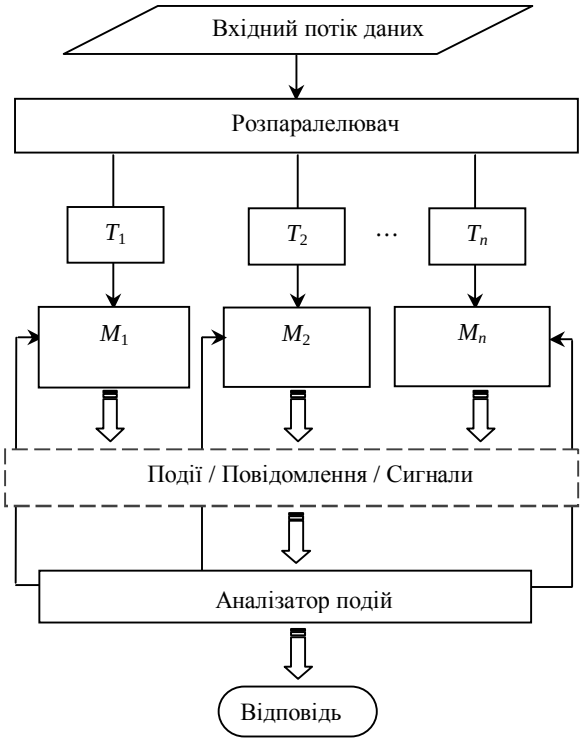

Рис. 1. Загальна схема системи

Для опису взаємодії аналізатора подій та інших процесів скористаємося позначеннями >>, *( )*, *[ ]*, <=. Вони мають такий зміст.

>> – операція, що забезпечує передачу значень від першого складового процесу до другого. Спосіб передачі не уточнюється й може бути довільний: через файл, пайп, udp- або tcp-потік, чи навіть у спосіб, досі невідомий.

[ ] – процес, вказаний у цих дужках, може бути відсутнім. Операція вводиться, оскільки деякі елементи наведеної схеми можуть бути відсутні.

<= – процес, вказаний ліворуч, має двосторонній обмін значеннями з елементами керування процесу, вказаного праворуч. Реалізація цієї операції вимагає наявності двосторонніх потоків значень або можливості двостороннього обміну сигналами чи повідомленнями.

( ) – цим фіксується, що є віртуальний процес, утворений процесами, вказаними в цих дужках.

Необхідно зауважити, що вхідний потік значень не є простим. Значення можуть надходити з різноманітних апаратних пристроїв або сховищ і баз даних, що зберігають потрібну інформацію. Ці частини утворюють множину процесів, які позначимо D_1, D_2, …, D_s. Розпаралелювач реалізується множиною відповідних процесів P_1, P_2, …, P_s, де кожен P_i забезпечує розпаралелювання значень, породжених процесом D_i. Отже, в системі можуть відбуватися віртуальні процеси $(D_i>>P_i)$, $i=1, …, s$.

Перетворювачі T_{ji}, де $j=1, …, n_i$, є необов'язковими. Вони поєднуються з функціональними модулями M_{ji}: $[T_{ji}]>>M_{ji}$. Поєднання процесу отримання вхідних даних $(D_i>>P_i)$ та процесів передачі даних на модулі утворює множину процесів $((D_i>>P_i)>>([T_{ji}]>>M_{ji}))$, $i=1, …, s$, $j=1, …, n_i$.

Аналізатор подій A приймає рішення про розпізнавання інформаційної одиниці вхідних даних та узгоджує різнорідну поведінку функціональних модулів. Взаємодія процесу в аналізаторі подій і процесів у функціональних модулях під час обробки вхідних даних від процесу D_i описується виразами $((D_i>>P_i)>>([T_{ji}]>>M_{ji}))<=A$, $i=1, …, s$, $j=1, …, n_i$.

5. Висновки та подальші напрямки роботи

В результаті проведених досліджень було запропоновано розширення поняття віртуального процесу та описано віртуальний процес системи інтерфейсу людини з комп'ютером через засоби жестового мовлення, що відкриває шлях до реалізації такої системи.

Подальші дослідження спрямовані на реалізацію модуля керування та узгодження запропонованих функціональних модулів і створення загальної системи розпізнавання жестової мови та дослідження її роботи.

Список використаних джерел

1. *von Neumann J.* First Draft of a Report on the EDVAC/ *J. von Neumann//*Moore School of Electrical Engineering, University of Pennsylvania. – 1945. – p.49.
2. *Brugali D.* Software Development: Case Studies in Java /D. Brugali, M. Torchiano // Addisson-Wesley. – 2005.
3. Garlan, D. Using Style to Give Meaning to Software Architecture / Abowd G., Allen R., Garlan D.// SIGSOFT '93: Foundations of Software Engineering, Software Engineering Notes, ACM Press. – Los Angeles, USA. – 1993. – Proceedings.– 1993. – Vol. 118. – No. 3. – pp. 9-20.
4. *Крак Ю.В.*, Коваль Ю.В., Тернов А.С. До розробки інтерактивного інтерфейсу моделювання та розпізнавання жестової інформації / Ю.В.Крак, Коваль Ю.В., А.С.Тернов //Вісник Київського національного університету імені Тараса Шевченка Серія фізико-математичні науки. – 2014. – №4. - С.175-178.

References

1. von NEUMANN J. (1945) *First Draft of a Report on the EDVAC.* Moore School of Electrical Engineering, University of Pennsylvania.
2. BRUGALI D., TORCHIANO M. (2005) *Software Development: Case Studies in Java* Addisson-Wesley.
3. ABOWD G., ALLEN R., GARLAN, D. (1993) *Using Style to Give Meaning to Software Architecture.* Los Angeles, USA. Proc. of SIGSOFT '93: Foundations of Software Engineering. Software Engineering Notes, ACM Press. Vol. 118, No.3, pp. 9–20.
4. KRAK IU.V., KOVAL IU.V., TERNOV A.S. (2014) *To creation of interactive interface for sign information modeling and recognition.* Bulletin of Taras Shevchenko National University of Kyiv Series Physics & Mathematics. Vol. 4. p.175-178.

Надійшла до редколегії 15.02.15