

РОЗРОБКА ТА ДОСЛІДЖЕННЯ СПЕЦІАЛІЗОВАНОЇ ОБОЛОНКИ ПІД ОПЕРАЦІЙНУ СИСТЕМУ LINUX

Горелова М.А., Плаксієв Ю.А.

***Національний технічний університет
«Харківський політехнічний інститут», м. Харків***

Розроблений оригінальний програмний продукт у вигляді спеціалізованої оболонки, призначений для обмеження функціоналу працівника на робочому місці для операційної системи Linux.

При створенні спеціалізованої оболонки використано мову програмування C#, стандартні бібліотеки (.NET, Threading, Linq, Text, Configuration, Xml та інші) та бібліотеки для створення інтерфейсу C# Forms.

Програмний продукт працює з певним списком програм, які адміністратор попередньо завантажує і налаштовує для використання працівником (оператором або технічним користувачем) на робочому місці. Після запуску спеціалізованої оболонки користувачу треба авторизуватись. Після успішної авторизації на екрані випадає список програм, які дозволені для використання. Їх можна запустити шляхом натискання на іконки програм, що відображаються на панелі задач.

Розроблено та досліджено багатопотоковість роботи програмного продукту, зручний для користувача інтерфейс, шифрування даних для авторизації і передачі даних між програмами. Проведена оптимізація роботи запитів передачі інформації між базою даних і програмою.

Для реалізації багатопотоковості роботи програми використано мову програмування C# і алгоритм розпаралелювання процесів та програм. Також було досліджено рівні програмних засобів. В програмі було розглянуто такі рівні як: мікропрограмні засоби, операційна система, інструментальні програми, програми користувача. Для захисту даних користувача в програмі реалізовано криптографічний алгоритм RSA із відкритим ключем. Для зберігання даних користувача використано об'єктно-реляційну систему управління базами даних PostgreSQL. Оптимізація передбачає використання конкретних імен стовпців, зведення до мінімуму використання підзапитів, використання індексації, обмеження розміру набору робочих даних та інше.

Інтерфейс програми реалізовано за допомогою бібліотеки C#Forms.

Проведено тестування програмного продукту. На кожному етапі розробки було використане модульне тестування, інтеграційне, системне і приймальне тестування.

Архітектура і відкритий код програмного забезпечення допускають подальшу модернізацію розробленого програмного продукту.