

Д. Н. Семененко, Е. В. Рафалова
(ГГУ имени Ф. Скорины, Гомель)

ВИРТУАЛЬНЫЕ МАШИНЫ И ЭМУЛЯЦИЯ: ЗАПУСК WINDOWS-ПРИЛОЖЕНИЙ НА LINUX

Многие образовательные программы и специализированное ПО (например, инженерные пакеты, Microsoft Office) доступны только для Windows, что создает сложности для пользователей Linux. Для решения этой задачи используются три ключевых подхода: VirtualBox [1] (полная виртуализация), Wine [2] (слой совместимости) и Docker [3] (контейнеризация). Каждый метод подходит для разных сценариев и имеет свои технические особенности. Но не во всех случаях требуется прибегать к таким методам, ведь некоторые дистрибутивы Linux, как Linux Mint, поставляются с нативными офисными пакетами, позволяющими обойтись без установки Windows-версий программ.

VirtualBox обеспечивает полную виртуализацию Windows в изолированной среде. После установки через терминал (`sudo apt install virtualbox` для Debian/Ubuntu) и загрузки образа Windows пользователь получает доступ к любой Windows-программе. Главное достоинство – почти полная совместимость, включая ресурсоемкие приложения вроде AutoCAD или MATLAB. Однако метод требует значительных ресурсов: минимум 4 ГБ оперативной памяти и выделение места на диске под виртуальную машину. Дополнительные сложности включают настройку общих папок и проброс USB-устройств.

Wine – слой совместимости, преобразующий вызовы Windows API в Linux-аналоги. Установка выполняется командой `sudo apt install wine`, а управление упрощается через графические оболочки вроде PlayOnLinux. Wine позволяет запускать .exe-файлы напрямую, без виртуализации, что обеспечивает высокую производительность. Например, Notepad++ или Excel 2010 работают почти нативно. Однако поддержка современных приложений (Office 365, игр с DirectX 12) ограничена, а для настройки зависимостей часто требуется ручное вмешательство через winetricks.

Docker контейнеризует приложения, изолируя их зависимости. С применением предварительно настроенных образов с Wine (например, jess/wine) можно запускать Windows-приложения в изолированном окружении. Пример команды для проброса графики выглядит следующим образом: `docker run -it --rm -e DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix jess/wineprogram.exe`. Контейнеризация обеспечивает переносимость и строгий контроль версий библиотек, что актуально для воспроизведения критически важного окружения (например, в академических исследованиях или при работе с legacy-кодом). В то же время данный метод требует дополнительной настройки X11-сервера (например, командой `xhost +local:docker`) и менее пригоден для мультимедийных приложений.

Запуск Windows-приложений на Linux остается актуальной задачей из-за ограниченной доступности специализированного ПО. VirtualBox идеален для полной изоляции и совместимости, но требует значительных ресурсов. Wine обеспечивает высокую производительность для старых приложений, однако его настройка зачастую нетривиальна. Docker предоставляет баланс между изоляцией и гибкостью, особенно для академических и legacy-проектов. Основным критерием выбора выступает поиск оптимального баланса между производительностью системы, совместимостью компонентов и затратами ресурсов. При этом развитие нативных альтернатив (например, офисных пакетов в Linux Mint) и технологий вроде Proton для игр в перспективе должно снизить зависимость от функционала Windows.

Литература

1. VirtualBox [Электронный ресурс]: Oracle VM VirtualBox User Manual. – Режим доступа : <https://www.virtualbox.org/manual/UserManual.html>. – Дата доступа: 20.03.2025.

2. GitLab [Электронный ресурс]: Wine User's Guide. – Режим доступа : <https://gitlab.winehq.org/wine/wine/-/wikis/Wine-User's-Guide>. – Дата доступа : 22.03.2025.
3. Docker [Электронный ресурс]: Docker Manuals. – Режим доступа : <https://docs.docker.com/manuals/>. – Дата доступа : 24.03.2025.