

### 3.3 МНОЖИННІ ПРИКЛАДНІ СЕРЕДОВИЩА

При реалізації множинних прикладних середовищ розробники стикаються з суперечливими вимогами. З одного боку, завданням кожного прикладного середовища є виконання програми по можливості так, як коли б вона виконувалася на «рідній» ОС. Але потреби цих програм можуть входити в конфлікт з конструкцією сучасної операційної системи. Спеціалізовані драйвери пристроїв можуть суперечити вимогам безпеки. Можуть конфліктувати схеми управління пам'яттю і віконні системи. Але найбільшою потенційною проблемою є продуктивність – прикладне середовище повинне виконувати програми з прийнятною швидкістю.

Цій вимозі не відповідають системи емуляції. Для скорочення часу на виконання «чужих» програм прикладні середовища використовують імітацію програм на рівні бібліотек. Ефективність цього підходу пов'язана з тим, що більшість сьогоднішніх програм виконуються під управлінням GUI (графічних інтерфейсів користувача) типу Windows, Mac або UNIX Motif. При цьому додатки витрачають велику частину часу, роблячи деякі добре передбачувані речі. Вони безперервно виконують виклики бібліотек GUI для маніпулювання вікнами і для інших пов'язаних з GUI дій. І це те, що дозволяє прикладним середовищам відшкодувати час, витрачений на емуляцію команди за командою. Модульність операційних систем нового покоління дозволяє набагато легше реалізувати підтримку множинних прикладних середовищ. На відміну від старих операційних систем, що складаються з одного великого блоку для усіх практичних застосувань, розбитого довільним чином на частини, нові системи є модульними, з чітко певними інтерфейсами між складовими. Це робить створення додаткових модулів, що об'єднують емуляцію процесора і трансляцію бібліотек, значно простішим.

До вдосконалених операційних систем, що явно містять засоби множинних прикладних середовищ, належать: IBM OS/2 2.x, Microsoft Windows NT і версії UNIX від Sun Microsystems, IBM і Hewlett-Packard.