

УДК 004.415.2:004.451.(045)

Черниш Л.Г. к.т.н.
Сягровець К.В.

ФУНКЦІОНУВАННЯ ПРИКЛАДНИХ ПРОГРАМ НА БАЗІ МІЖНАРОДНОГО СТАНДАРТУ POSIX

Національний авіаційний університет, e-mail: deniski@ukr.net

Мобільність програмного забезпечення - це дуже важлива і складна сучасна задача. Виходячи з того, що стандартизація оточення додатків є одним з методів підвищення мобільності, у статті проведено аналіз основних понять і структури стандарту POSIX з точки зору функціонування прикладних програм. Розглянуто середу компіляції і мобільність додатків, сумісних з POSIX.

Ключові слова: стандарт, додаток, POSIX, мобільність, сервіс, середа компіляції.

Постановка завдання дослідження. Забезпечення мобільності (портабельності) програмного забезпечення (ПО) - завдання виняткової важливості і складності; у наш час ця обставина навряд чи потребує просторових обґрунтувань. Один із загальноприйнятих способів підвищення мобільності ПЗ - стандартизація оточення додатків : програмних інтерфейсів, що надаються, утиліт і тому подібне. На рівні системних сервісів подібне оточення описує стандарт POSIX (Portable Operating System Interface - мобільний інтерфейс операційної системи); назва запропонована відомим фахівцем, засновником Фонду вільного програмного забезпечення Р. Столманом. Найбільш сучасна з доступних версій стандарту POSIX, в редакції 2003 р., яку можна назвати "стандартом втройне", а саме: стандартом IEEE Std 1003.1, Технічним стандартом Open Group [6] і, що найважливіше, міжнародним стандартом ISO/IEC 9945 [1-4].

Аналіз існуючих досліджень і постановка завдання. На початку 1998 р. представники трьох організацій - Комітету із стандартів мобільних додатків Інституту інженерів по електротехніці і електроніці, Open Group і робочої групи 15 підкомітету 22 спільні технічні комітети 1(JTC1/SC22/WG15) Міжнародної організації із стандартизації - почали консультації з питання злиття і розвитку курируваних ними стандартів інтерфейсів до системних сервісів: IEEE Std 1003.1, IEEE Std 1003.2, базовий специфікація від Open Group, ISO/IEC 9945-1, ISO/IEC 9945-2. У вересні того ж року в Остіні, штат Техас, в офісі корпорації IBM відбулося організаційне засідання групи, сформованої для досягнення поставленої мети [8]. Засадничим документом для переглянутого стандарту, перший проект якого був представлений в липні 1999 року, стали Базові специфікації від Open Group, оскільки вони включали положення стандартів IEEE і ISO/IEC. У 2001 році, після закінчення підготовчої роботи, стандарт містив наступні чотири частини: основні визначення; опис прикладного програмного С- інтерфейсу до системних сервісів; опис інтерфейсу до системних сервісів на рівні командної мови і службових програм; детальне роз'яснення положень стандарту, обґрунтування прийнятих рішень. Далі в ISO, IEEE і Open Group з більшою або меншою швидкістю (у 2001-2002 рр.) пройшло формальне затвердження нового стандарту POSIX. Тим часом накопичувалися відносно дрібні виправлення, враховані в редакції 2003-го року. З розвитком стандарту розширювалося і трактування терміну "POSIX". Спочатку він відносився до документу IEEE Std 1003.1-1988, що описував прикладний програмний інтерфейс ОС класу Unix. Після стандартизації інтерфейсу на рівні командної мови і службових програм правильніше розуміти під словом "POSIX" стандарт в цілому, означаючи перелічені вище частини 2 і 3 через POSIX.1 і POSIX.2 відповідно до нумерації документів IEEE і ISO/IEC. **Метою роботи** є аналіз основних понять і структури стандарту POSIX з точки зору функціонування застосованих програм.

Основна частина дослідження. Стандарт POSIX описує безліч базових, системних сервісів, необхідних для функціонування застосованих програм. Доступ до них надається за допомогою інтерфейсу, специфікованого для мови С, командної мови і загальноновживаних службових програм. У кожного інтерфейсу є дві сторони: що викликає і викликається. Стандарт POSIX орієнтований в першу чергу на ту, що викликає. Його мета - зробити додатки мобільними на рівні початкової мови. Це означає, зокрема, що при перенесенні С- програм на іншу операційну платформу знадобиться перекомпіляція. Про мобільність здійснених програм і/або об'єктних файлів не йдеться. Стандарт POSIX зовсім не обмежений рамками Unix- середовища. Існують операційні системи (ОС) "Незалежного походження", що надають необхідні сервіси і тим самим що підтримують виконання POSIX- сумісних застосувань. Можна стверджувати, що наслідування стандарту POSIX полегшує перенесення додатків практично на будь-яку скільки-небудь поширену операційну платформу. Додаткові зусилля із підвищення мобільності, ті, що додаються на етапі розробки, безумовно, окупляться.

Визначаючи інтерфейс до системних сервісів, POSIX залишає за рамками розгляду їх реалізацію.

Зокрема, не розрізняються системні виклики і бібліотечні функції. Не є об'єктом стандартизації засобу адміністрування, апаратні обмеження і функції, необхідні тільки суперкористувачеві, що ще раз підкреслює спрямованість стандарту POSIX на додатки, а не на операційні системи. POSIX нейтральний по відношенню до системної архітектури і розрядності процесора. Це дуже важливий аспект мобільності додатків. Орієнтація на міжнародний стандарт мови C визначила не лише стиль опису функцій, але і, до деякої міри, напрям розвитку специфікацій POSIX в плані синхронізації обох стандартів. Як відомо в затвердженій в 1999 р. редакції специфікацій мови C [5] узаконений комплексний тип даних, що викликало відповідне поповнення POSIX- функцій. У стандарті POSIX проведено розділення на обов'язкові і додаткові функції, причому обов'язкове ядро зроблене по можливості компактним. Зрозуміло, особлива увага приділяється способам реалізації стандартизованих функцій як в "класичному" Unix- середовищі, так і на інших операційних платформах, в мережесхемних і розподілених конфігураціях. Розробники нової версії стандарту POSIX дуже дбайливо віднеслися і до його передісторії, і до передісторії Unix- систем, і, головне, до додатків, задовольняючих більше ранніх версій стандарту. Існуючі інтерфейси намагалися зберегти; в процесі розвитку дотримувалася принцип зворотної сумісності; нові інтерфейси додавалися так, щоб вони не конфліктували із старими. Повністю уникнути внесення змін до додатків не вдалося з цілком зрозумілих причин: потрібно було усунути протиріччя між різними початковими специфікаціями, а також відмовитися від підтримки "традиційного" варіанту мови C і перейти на його міжнародний стандарт.

Основні поняття стандарту. Стандарт POSIX в редакції 2003-го року - дуже великий, багатогранний документ, де детально розглядаються наступні категорії системних компонентів :

- ❖ засоби розробки;
- ❖ мережесхемні засоби;
- ❖ засоби реального часу;
- ❖ потоки управління;
- ❖ математичні інтерфейси;
- ❖ пакетні сервіси;
- ❖ заголовні файли;
- ❖ успадковані інтерфейси.

Саме такий (на верхньому рівні, далеко не повний) репертуар повинна надавати операційна система для роботи додатка. Найважливішим є поняття відповідності стандарту POSIX. Ми вже відмічали, що всякий інтерфейс має в розпорядженні двома сторонами: що викликає і викликається. Дві сторони є і у POSIX- відповідності : відповідність реалізації і додатка. Реалізація (операційна система), що відповідає стандарту POSIX, повинна підтримувати усі обов'язкові службові програми, функції, заголовні файли із забезпеченням специфікованої в стандарті поведінки. Константа `_POSIX _VERSION` має значення 200112L. ОС може надавати можливості, помічені в стандарті в якості додаткових, а також містити нестандартні функції. Якщо стверджується, що підтримується деяке розширення, це повинно робитися несуперечливим чином, для усіх необхідних частин і так, як описано в стандарті. У заголовному файлі `<unistd.h>` слід визначити константи, що відповідають підтримуваним необов'язковим можливостям (наприклад, константа `_POSIX2 _C _DEV` обслуговує засоби розробки на мові C). Аналізуючи ці константи під час компіляції, додаток з'ясує можливості використовуваної ОС і підлаштується під них. Аналогічні дії на етапі виконання можуть бути виконані за допомогою функції `sysconf ()` і/або службової програми `getconf`.

Для мінімізації розмірів ОС і додатків стандартом POSIX передбачена дуже дрібна гранулярна необов'язкових можливостей (всього їх сорок). З іншого боку, проведено об'єднання взаємозв'язаних необов'язкових можливостей в групи, що у багатьох випадках позбавляє від аналізу великого числа опцій. Групи ці такі: шифрування; засоби реального часу; просунуті засоби реального часу; потоки реального часу; просунуті потоки реального часу; трасування; потоки; успадковані можливості.

Наприклад, до групи "Засоби реального часу" (`_XOPEN _REALTIME`) входять можливості чотирнадцяти видів, у тому числі планування на основі пріоритетів, асинхронне введення/виведення, семафори, таймери і тому подібне. Як приклад, розглянемо результат застосування утиліти `getconf` до однієї з версій ОС Linux :

```
$ getconf _POSIX_VERSION
199506
$ getconf POSIX2_C_DEV
1
$ getconf _XOPEN_REALTIME
1
$ getconf _POSIX_TRACE
```

undefined

Це означає, що підтримується застаріла версія стандарту POSIX, серед інших є присутніми засоби розробки, і можливості реального часу; засоби трасування відсутні. У документації на ОС мають бути відбиті питання відповідності стандарту POSIX, описані підтримувані додаткові і нестандартні можливості. Для додатків поняття відповідності стандарту POSIX багатіше нюансами. Передбачена строга відповідність, головна відмінна ознака якої - обмеження круга використовуваних можливостей рамками стандарту. Розглядається і відповідність із застосуванням розширень; в цьому випадку документація на додаток повинна містити опис необхідних нестандартних можливостей. Бажано, щоб використовувані розширення POSIX- можливостей описувалися міжнародними і/або національними стандартами. Відмітимо, що для реалізації поняття строгої POSIX- відповідності безглузде хоч би з тієї причини, що не буває операційних систем без засобів адміністрування, а вони не описуються цим стандартом.

Допускається існування "підпрофіль", що описують підмножини стандартних можливостей. Реалізація, що відповідає підпрофілью, може функціонувати на апаратних платформах з обмеженими ресурсами і/або обслуговувати потреби специфічних застосувань. До числа найважливіших належать поняття, що описують поведінку реалізації в різних ситуаціях. Для багатьох коректних ситуацій поведінка буває неспецифікованою, тобто мобільне застосування не повинне покладатися на збіг поведінки різних реалізацій. Для некоректних ситуацій поведінка може бути невизначеною; додатку не лише не слід покладатися на певний характер подібної поведінки - воно не повинне здійснювати некоректних дій, що викликають невизначену поведінку. Ще один близький термін, "поведінка, залежна від реалізації", додатково означає, що поведінку реалізації необхідно документувати. Стандарт POSIX - це існуючий багато років, організм, що розвивається, в якому з кожною новою редакцією щось з'являється, а щось втрачається. Застарілими називаються можливості, які ще підтримуються різними реалізаціями, але в майбутньому вони, ймовірно, відінуть. Нові застосування не повинні їх використати; для кожної з них стандартом передбачена адекватна по функціональності сучасна заміна. Більше обмежений сенс наданий терміну "успадкований": він описує застарілі необов'язкові можливості, яких, зрозуміло, слід уникати в нових застосуваннях.

Основні поняття операційних систем, що відповідають стандарту POSIX. Розглянемо наступні

основні поняття операційних систем, що відповідають стандарту POSIX :

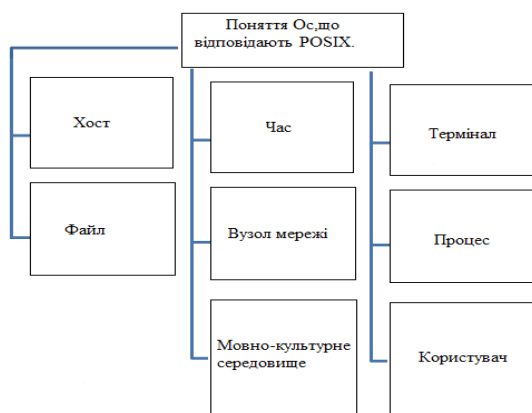


Рис.1 Схема понять ОС, що відповідають POSIX.

Це первинні поняття. Їх не можна строго визначити, але можна пояснити за допомогою інших понять і стосунків. Для кожного з виділених понять будуть описані властиві їм атрибути і застосовані до них операції. У тексті стандарту POSIX містяться наступні пояснення основних понять разом з посиланнями на атрибути і операції :

1. У користувача є ім'я і числовий ідентифікатор.

2. Файл - об'єкт, що допускає читання і/або запис і має такі атрибути, як права доступу і тип. До числа останніх відносяться звичайний файл, символьний і блочний спеціальні файли, канал, символьне посилання, сокет і каталог. Реалізація може підтримувати і інші типи файлів.

3. Процес - адресний простір разом з виконуваними в нім потоками управління, а також системними ресурсами, які цим потокам потрібно.

4. Термінал (чи термінальний пристрій) - символьний спеціальний файл, що підкоряється специфікаціям загального термінального інтерфейсу.

5. Мережа - сукупність взаємозв'язаних хостів.

6. Мовно-культурне середовище - частина призначеного для користувача оточення, залежна від мовних і культурних угод. Механізм прав доступу дозволяє вибірково дозволяти і забороняти здійснення подібних операцій. Раніше в стандарті фігурувало поняття суперкористувача, не схильного до контролю прав доступу. У POSIX - 2001 вибрано гнучкіше формулювання - що "має відповідні привілеї", що відбиває прогрес в реалізації ОС з розщеплюванням суперпризначених для користувача можливостей. У POSIX- сумісних ОС визначені об'єкти, які можна назвати допоміжними; вони допомагають організувати взаємодію між основними сутностями. Особливо широкий спектр засобів межпроцесного взаємодії. Процеси виконуються в певному оточенні, частиною якого є мовно-культурне середовище (Locale), утворене такими категоріями, як символи і їх властивості, формати повідомлень, дата і час, числові і

грошові величини. Як правило, з процесом асоційовані принаймні три файли - стандартне введення, стандартне виведення, стандартний протокол. Зазвичай стандартне введення призначається на клавіатуру терміналу, а стандартне виведення і стандартний протокол - на екран. Із стандартного введення читаються команди і (іноді) початкові дані для них. На стандартне виведення поступають результати виконання команд. У стандартний протокол поміщаються діагностичні повідомлення. До операційних систем можуть пред'являтися якісні вимоги, наприклад, вимога підтримки реального часу : здатність забезпечити необхідний сервіс впродовж заданого відрізка часу.

Середовище компіляції POSIX- сумісних застосувань. Як правило, розробка додатків ведеться в кросс-режиме, тобто платформа розробки не співпадає з платформою виконання. На інструментальній платформі створюється середовище компіляції додатків, так що результат компіляції може бути перенесений для подальшого виконання на цільову платформу. Найважливіша частина середовища компіляції - заголовні (чи що включаються) файли, що містять прототипи функцій, определения символических констант, макросів, типів даних, структур і тому подібне. Для кожної описаної в стандарті POSIX функції визначено, які заголовні файли мають бути включені застосуванням, що використовує її. Вище було вказано, що за допомогою символічних констант, визначених в заголовному файлі <unistd.h>, операційна система надає додатку інформацію про підтримувані можливості. Стандартом POSIX передбачений симетричний механізм, що називається механізмом макросів перевірки можливостей, він дозволяє додаткам оголошувати про своє бажання отримати доступ до певних прототипів і імен.

Мобільність POSIX- сумісних застосувань. Мобільність додатків, що відповідають стандарту POSIX, принципово досяжна завдяки двом основним чинникам. По-перше - ця наявність величезного числа стандартизованих системних сервісів, а по-друге - можливість динамічного з'ясування характеристик цільової платформи і підстроювання під них додатка (маємо на увазі мобільність у рамках, що регламентуються стандартом). Додатки, що відповідають стандарту POSIX, можуть бути одно- і многопроцесними, з можливістю динамічної адаптації конфігурації до властивостей цільової платформи. Стандартизовані засоби породження і завершення процесів, зміни їх програм, опитування і/або зміни різноманітних характеристик.

У стандарті POSIX ретельно пропрацювали питання доступу до зовнішніх пристроїв, приєднаних по послідовних лініях, особливо до терміналів. Можливо, більшої деталізації потребують засоби роботи з такими поширеними носіями, як магнітна стрічка.

Стандартизований командна мова shell - адекватний засіб для написання невеликих мобільних процедур і їх швидкої інтерактивної відладки. Виділимо механізм конвеєрів, що дозволяє об'єднувати команди в ланцюжки з фільтрацією проміжних результатів. Службові програми утворюють розвинене середовище виконання для shell- процедур. За рахунок фонового режиму можна організувати одночасне виконання декількох програм і взаємодію з ними за допомогою звичайного терміналу без багатовіконних можливостей (втім, вікна, поза сумнівом, не завадили б). POSIX стандартизує інтерфейс командного рядка. В принципі, він достатній, в міру зручний і, що важливо, створює мінімум проблем з точки зору мобільності. Ймовірно, в майбутніх версіях стандарту буде регламентований графічний інтерфейс, але, безумовно, це відгукнеться додатковими складнощами для розробників мобільних застосувань. Мовно-культурне середовище - одно з найважливіших понять стандарту POSIX з точки зору мобільності. Додатки здатні визначати потрібне їм середовище і адаптуватися до потреб користувачів. Для розрахованих на багато користувачів систем потрібно організацію взаємодії великого числа людей. POSIX вирішує цю проблему, регламентуючи засоби безпосереднього і поштового обміну інформацією. Стандартом POSIX передбачені базові засоби підтримки розробки, що, звичайно, не знижує потреби в спеціалізованих, розвинених системах, коли йдеться про роботу з дійсно великими програмними проектами. Додаткам надаються стандартизовані засоби для з'ясування як "великоблочних" характеристик цільової системи, так і дрібніших характеристик.

Висновки. Таким чином, в цій статті проведений аналіз основних понять і структури стандарту POSIX з точки зору функціонування застосованих програм. Розглянуто середовище компіляції і мобільність додатків, сумісних з POSIX.

Список використаних джерел:

1. Information technology - Portable Operating System Interface (POSIX) - Part 1: Base Definitions ISO/IEC 9945-1:2003
2. Information technology - Portable Operating System Interface (POSIX) - Part 2: System Interface ISO/IEC 9945-2:2003
3. Information technology - Portable Operating System Interface (POSIX) - Part 3: Shell and Utilities ISO/IEC 9945-3:2003
4. Information technology - Portable Operating System Interface (POSIX) - Part 4: Rationale ISO/IEC 9945-4:2003
5. Галатенко А.В. О скрытых каналах и не только Jet Info, 2002, 11
6. The Open Group Base Specifications Issue 6 IEEE Std 1003.1, 2003 Edition. - The IEEE and The Open Group, 2001-2003