1. Знайомство з операційною системою Linux.1.1. Вступ Всі матеріали першої і наступної тем будемо розглядати на прикладах,пов'язаних з використанням однієї з різновидів операційної системи UNIX –операційної системи Linux. Для забезпечення безпеки системи пропонується розпочати знайомствоз операційною системою Linux в середовищі віртуального комп'ютера,організованого засобами системи керування віртуальними комп'ютерамиVirtualBox (http://www.virtualbox.org).В якості дистрибутива Linux рекомендується обрати дистрибутивDebian GNU Linux (http://www.debian.org) або користуватись LiveDVDKnoppix (http://www.knoppix.net).Перший варіант вимагає інсталяції на віртуальний комп'ютер, а привиборі другого — користувач може одразу після завантаження віртуальногокомп'ютера працювати з системою Linux.Однак особливістю роботи в LiveDVD Knoppix, як і більшості Live-систем при використанні налаштувань за замовчуванням, є втрата усіхкористувацьких змін у системі після перезавантаження комп'ютера, в томучислі і створених під час роботи документів. Тому важливо слідкувати за тим,де будуть зберігатись створені документи. Найкраще їх зберігати нажорсткому диску (у випадку роботи з віртуальним комп'ютером його потрібностворити в середовищі віртуальних комп'ютерів і потім розбити на розділи тавідформатувати), в мережній папці або на флеш-носії. У LiveDVD Knoppix єможливість створити спеціальний файл на жорсткому диску, де будутьзберігатись усі зміни, зроблені користувачем під час роботи в системі(встановлені нові програми, налаштування програм, нові документи). У версії Linux від Knoppix робота в системі за замовчуваннямздійснюється під користувачем knoppix. Можна створити також свогокористувача. Якщо є потреба виконати команду з правами адміністраторасистеми (root), користуються командою sudosudo lsПісля виконання команди, вказаної у sudo, інші команди будутьвиконуватися з правами звичайного користувача.При інсталяції системи обов'язково слід запам'ятати парольсуперкористувача (користувача з іменем root), а також інших користувачів,які створювались у системі під час інсталяції.У тексті практичних занять програмні конструкції, включаючи іменасистемних викликів, стандартних функцій і команди оболонки операційноїсистеми, виділені іншими шрифтами. У Linux системні виклики і командиоболонки ініціюють складні послідовності дій, які стосуються різних аспектівфункціонування операційної системи. Як правило, у рамках однієї темипрактичних занять повне пояснення всіх особливостей їхньої поведінки є

неможливим. Тому детальні описи більшості системних викликів, системнихфункцій і деяких команд оболонки операційної системи при першій зустрічі зними винесені з основного тексту і виділені двома вертикальними рисками, ав основному тексті розглядаються тільки ті деталі їхнього опису, длярозуміння яких вистачає набутих знань. Крім того, винесені описи не єповним описом команд, а адаптованим стосовно даного курсу. Для одержанняповного опису звертайтеся до Linux Manual.Якщо деякий параметр у команди оболонки є необов'язковим, він будевказуватися у квадратних дужках, наприклад, [who]. У випадку, колиможливий вибір тільки одного з декількох можливих варіантів параметрів,варіанти будуть перераховуватися у фігурних дужках і розділятисявертикальною рискою, наприклад, {+|-|=}.1.2. Коротка історія операційної системи Linux, їїструктураЯдро операційної системи Linux є монолітною системою. Прикомпіляції ядра Linux можна дозволити динамічне завантаження івивантаження дуже багатьох компонентів ядра – так званих модулів. У моментзавантаження модуля його код завантажується для виконання впривілейованому режимі і зв'язується з іншою частиною ядра. Всерединімодуля можуть використовуватися будь-які експортовані ядром функції.Свого нинішнього виду ця операційна система набула в результатітривалої еволюції UNIX-подібних операційних систем. Історія розвитку UNIXдетально освітлена практично у всій літературі, присвяченій обчислювальнійтехніці. Досить детально викладено історію у книзі [8] або в оригінальнійроботі одного з родоначальників UNIX [18]. Для нас найбільш важливим єіснування двох базових ліній еволюції – лінії System V і лінії BSD, оскільки впроцесі навчання ми будемо зіштовхуватися з розходженнями в їхнійреалізації.1.3. Системні виклики і бібліотека libcОсновною постійно функціонуючою частиною операційної системиLinux є її ядро. Інші програми (системні або користувацькі) можутьспілкуватися з ядром за допомогою системних викликів, які фактично єпрямими точками входу програм у ядро. При виконанні системного викликупрограма користувача тимчасово переходить у привілейований режим,одержуючи доступ до даних або пристроїв, які недоступні при роботі врежимі користувача. Реальні машинні команди, необхідні для активізації системнихвикликів, природно, відрізняються між комп'ютерами, як і спосіб передачіпараметрів та результатів між програмою і ядром. Однак з поглядупрограмування мовою C використання системних викликів нічим зовні невідрізняється від використання інших функцій стандартної ANSI бібліотекимови C, таких, наприклад, як функції роботи з рядками strlen(),

strcpy(). Стандартна бібліотека Linux – libc – забезпечує C-інтерфейс докожного системного виклику. Це призводить до того, що системний викликвиглядає як функція мовою C для програміста. Крім того, багато із ужевідомих Вам стандартних функцій, наприклад функції для роботи з файлами:fopen(), fread(), fwrite() при реалізації в операційній системі Linuxбудуть використовувати різні системні виклики. Під час вивчення курсупознайомимося з великою кількістю різноманітних системних викликів таїхніх C-інтерфейсів. Більшість системних викликів, що повертають ціле значення,використовують значення -1 для повідомлення про виникнення помилки, азначення більше або рівне 0 – при нормальному завершенні. Системнівиклики, що повертають покажчики, зазвичай для ідентифікації помилковоїситуації використовують значення NULL. Для точного визначення причинипомилки C-інтерфейс надає глобальну змінну errno, яка описана у файлі<errno.h> з її можливими значеннями і їхніми короткими визначеннями.Відмітимо, що аналізувати значення змінної errno необхідно одразу ж післявиникнення помилкової ситуації, тому що системні виклики, які успішнозавершилися, не змінюють її значення. Для одержання символьної інформаціїпро помилку на стандартному виводі програми для помилок (зазамовчуванням екран термінала) може застосовуватися стандартна UNIX-функція perror().Функція perror()Прототип функції#include <stdio.h>void perror(char \*str);Опис функціїФункція perror() призначена для виводу повідомлення пропомилку, що відповідає значенню системної змінної errno в стандартнийпотік виводу помилок. Функція друкує вміст рядка str (якщо параметрstr не дорівнює NULL), двокрапку, пробіл і текст повідомлення, щовідповідає помилці, з наступним символом переходу на новий рядок ('\n').1.4. Вхід вопераційну систему Linux і зміна пароляОпераційна система Linux є багатокористувацькою операційноюсистемою. Для забезпечення безпечної роботи користувачів і цілісностісистеми, доступ до неї повинен бути санкціонований. Для кожногокористувача, якому дозволений вхід у систему, створюється спеціальнереєстраційне ім'я – username (абоlogin)і зберігається спеціальний пароль –password, що відповідає цьому імені. Як правило, при створенні новогокористувача початкове значення пароля для нього задає системнийадміністратор. Після першого входу в систему користувач повинен змінити

початкове значення пароля за допомогою спеціальної команди. Надалі вінможе в будь-який момент змінити пароль за своїм бажанням.Якщо в системі встановлена графічна оболонка поряд зі звичайнимиалфавітно-цифровими терміналами, найкраще здійснити перший вхід всистему з алфавітно-цифрового термінала або його емулятора. Перемкнутисяз графічної оболонки в консольний термінал можна за допомогою комбінаціїклавіш Ctrl+Alt+Fx, деFx – одна із функціональних клавіш F1, F2, ...Залежно від використовуваного дистрибутиву Linux, налаштувань системи, є 7алфавітно-цифрових терміналів, які відповідають функціональним клавішамF1, F2, ..., F7. Дуже часто перший термінал використовується длязавантаження системи і з нього завантажується графічна оболонка. Дляперемикання між алфавітно-цифровими терміналами достатньо комбінаціїклавіш Alt+Fx. Для повернення до графічної оболонки використовуютькомбінацію клавіш Alt+F7 або Alt+F8.Під час входу в систему з алфавітно-цифрового термінала на екраніз'являється напис, що пропонує ввести реєстраційне ім'я. Найчастіше це"login:". Набравши своє реєстраційне ім'я, натискають клавішу <Enter>.Система запитає пароль, який відповідає введеному імені, видавши спеціальнезапрошення – "Password:". Уважно наберіть пароль, встановлений для Вассистемним адміністратором, і натисніть клавішу <Enter>. Пароль, щовводиться, на екрані не відображається, тому набирайте його акуратно!Якщо все було зроблено правильно, у Вас на екрані з'явиться запрошення довводу команд операційної системи.Пароль, встановлений системним адміністратором, необхідно змінити.Найчастіше для цього використовується команда passwd. У більшості UNIX-подібних систем потрібно, щоб новий пароль мав не менше шести символів імістив, принаймні, дві не букви і дві не цифри. Придумайте новий пароль і добре його запам'ятайте, а кращезапишіть. Паролі в операційній системі зберігаються в закодованому виді, іякщо Ви його забули, ніхто не зможе допомогти Вам його згадати. Єдине, щоможе зробити системний адміністратор, так це встановити Вам новий пароль. Введіть команду для зміни пароля. Зазвичай система просить спочаткунабрати старий пароль, потім ввести новий і підтвердити правильність йогонабору повторним введенням. Після зміни пароля вже ніхто сторонній незможе ввійти в систему під вашим реєстраційним іменем.Тепер Ви повноцінний користувач операційної системи Linux. 1.5. Спрощене поняття про пристрій файлової системи вLinux. Повні та відносні імена файлівВ операційній системі Linux існують три базових поняття: "процес","файл" і "користувач". З поняттям "користувач" ми щойно ознайомилися ібудемо користуватися ним надалі при вивченні роботи цієї операційноїсистеми. Поняття "процес" характеризує динамічну сторону того, щовідбувається в обчислювальній системі. Поняття "файл" характеризує

статичну сторону обчислювальної системи.З попереднього досвіду роботи з обчислювальною технікою Ви вжемаєте уявлення про файл, як про іменований набір даних, що зберігається де-небудь на магнітних дисках або інших носіях. Нам досить такого розуміння,щоб розібратися в тому, як організована робота з файлами в операційнійсистемі Linux. Детальніший розгляд поняття "файл" і організації файловихсистем для операційних систем у цілому буде наведено в темі 7, якаприсвячена організації файлових систем в Linux.Всі файли, доступні в операційній системі Linux, як і у вже відомихВам операційних системах, об'єднуються в деревоподібну логічну структуру.Файли можуть об'єднуватися в каталоги або директорії. Не існує файлів, якіне входили б до складу якого-небудь каталогу. Каталоги можуть входити доскладу інших каталогів. Допускається існування порожніх каталогів, у які невходить жоден файл і жоден інший каталог (див. рис. 1.1). Серед всіх каталогівіснує тільки один каталог, що не входить до складу інших каталогів – йогоприйнято називати кореневий каталог (коренева директорія). Напочатковому рівні вивчення Linux ми можемо вважати, що у файловій системіLinux є, принаймні, два типи файлів: звичайні файли, які можуть містититексти програм, код для виконання, дані – їх ще прийнято називатирегулярними файлами, і каталоги. Кожен файл (регулярний або каталог) повинен мати ім'я. У різнихверсіях операційної системи UNIX існують певні обмеження на побудовуімені файлу. У стандарті POSIX на інтерфейс системних викликів дляопераційної системи UNIX міститься лише три явних обмеження: •Не можна створювати імена більшої довжини, ніж це передбаченоопераційною системою (для Linux – 255 символів). •Не можна використовувати символ NUL (не плутайте з покажчикомРис. 1.1. Приклад структури файлової системи

NULL!) – він же символ з нульовим кодом, він же ознака кінця рядкав мові C. •Не можна використовувати символ '/'. Також небажано застосовувати символи "зірочка" – "\*", "знакзапитання" – "?", "лапки" – """, "апостроф" – "'", "пробіл" – " " і "зворотнійслеш" – "\" (символи записані в стандарті запису символьних констант мовиC).Єдиним винятком є кореневий каталог, що завжди має ім'я "/". Цейкаталог є єдиним файлом, що повинен мати унікальне ім'я у всій файловійсистемі. Для всіх інших файлів імена повинні бути унікальними тільки врамках того каталогу, у який вони безпосередньо входять. Яким же чиномвідрізнити два файли з іменами "aaa.c", що входять у директорії "b" і "d"на рис. 1.1, щоб було зрозуміло про який з них йде мова? Тут на допомогуприходить поняттяповного імені файлу.Побудуємо шлях від кореневої вершини дерева файлів до файлу, щоцікавить нас, і випишемо всі імена вузлів дерева, що зустрічаються нанашому шляху, наприклад, "/ usr b aaa.c". У цій послідовності першимбуде завжди стояти ім'я кореневого каталогу, а останнім – ім'я файлу, щоцікавить нас. Відокремимо імена вузлів один від одного в цьому записі непробілами, а символами "/", за винятком імені кореневого каталогу інаступного імені каталогу ("/usr/b/aaa.c"). Отриманий запис однозначноідентифікує файл у всій логічній конструкції файлової системи. Такий запис іодержав назву повного імені файлу. 1.6. Поняття про поточнийкаталог. Команда pwd. Відносніімена файлів. Домашнійкаталог користувачаПовні імена файлів можуть містити в собі досить багато імен каталогіві бути дуже довгими, а тому з ними не завжди зручно працювати. У той жечас, існують такі поняття як поточний або робочий каталог і відносне ім'яфайлу. Для кожної програми, яка працює в операційній системі, включаючикомандний інтерпретатор (shell), що обробляє введені команди і виводитьзапрошення для їхнього введення, один з каталогів у логічній структуріфайлової системи призначається поточним або робочим для даної програми.Довідатися, який каталог є поточним для вашого командного інтерпретатора,можна за допомогою команди операційної системи pwd. Знаючи поточний каталог, ми можемо побудувати шлях по графуфайлів від поточного каталогу до файлу, що цікавить нас. Запишемопослідовність вузлів, які зустрінуться на цьому шляху, у такий спосіб. Вузол,що відповідає поточному каталогу, у запис не включаємо. При русі в напрямкудо кореневого каталогу кожний вузол будемо позначати двома символами"крапка" – "..", а при русі по напрямку від кореневого каталогу будемозаписувати ім'я вузла, що зустрівся. Розділимо позначення, що відносяться до

різних вузлів у цьому записі, символами "/". Отриманий рядок прийнятоназивати відносним іменем файлу. Відносні імена файлів змінюються призміні робочого каталогу. Так, у нашому прикладі, якщо робочий каталог – це"/d", то для файлу "/usr/b/aaa.c" відносним іменем буде"../usr/b/aaa.c", а якщо робочий каталог – це "/usr/b", то йоговідносне ім'я – "ааа.с".Для повноти картини ім'я поточного каталогу можна вставляти увідносне ім'я файлу, позначаючи поточний каталог одним символом "крапка" –".".Тоді наші відносні імена будуть виглядати як "./../usr/b/aaa.c" і"./aaa.c" відповідно.Програми, запущені за допомогою командного інтерпретатора, будутьмати як робочий каталог його робочий каталог, якщо всередині цих програмне змінити її розташування за допомогою спеціального системного виклику.Для кожного нового користувача в системі заводиться спеціальнийкаталог, що стає поточним одразу після його входу в систему. Цей каталогодержав назву домашнього каталогу користувача. Домашні каталоги користувачів знаходяться у каталозі /home.Наприклад, для користувача stud домашнім буде каталог /home/stud. Дляадміністратора (root) домашнім каталогом є /root. Синонімом домашньогокаталога є символ "~".1.7. Одержання допомоги про команди у ОС LinuxПід час вивчення операційної системи Linux Вам часто буде потрібнаінформація про те, для чого призначена команда або системний виклик, які вних параметри та опції, для чого призначені деякі системні файли, який їхнійформат. Більша частина інформації в UNIX Manual доступна в інтерактивномурежимі за допомогою утиліти man. Користуватися утилітою man досить просто – наберіть командуman ім'я де ім'я – це ім'я команди, що цікавить вас, утиліти, системного виклику,бібліотечної функції або файлу. Спробуйте з її допомогою подивитисяінформацію про команду pwd.Щоб пролистати сторінку отриманого опису, якщо він не помістився наекрані повністю, потрібно натиснути клавішу <пробіл>. Для прокручуванняодного рядка скористайтеся клавішею <Enter>. Повернутися на сторінкуназад дозволить одночасне натискання клавіш <Ctrl> і <b>. Вийти з режимуперегляду інформації можна за допомогою клавіші <q>.Іноді імена команд інтерпретатора і системних викликів або які-небудьще імена збігаються. Тоді, щоб знайти інформацію, необхідно задати утилітіman категорію, до якої належить ця інформація (номер розділу). Розподілінформації на категорії може трохи відрізнятися у різних версіях UNIX. Якщо Ви знаєте розділ, до якого відноситься інформація, то утилітуman можна викликати в Linux з додатковим параметром

man номер\_розділа ім'яВ інших операційних системах цей виклик може виглядати інакше. Дляодержання точної інформації про розбивку на розділи, форму вказання номерарозділу і додаткових можливостей утиліти man наберіть команду man manДля одержання допомоги можна використовувати також команду infoinfo lsЩоб отримати коротшу інформацію про командуls –infoОдержати підказку про команду можна за допомогою команди whatiswhatis ls1.8. Перегляд вмісту каталогу, зміна поточногокаталогуДля зміни поточного каталогу командного інтерпретатора можнаскористатися командою cd(change directory). Для цього необхіднонабрати команду у виглядіcd ім'я\_директоріїде ім'я\_директорії – повне або відносне ім'я каталогу, який потрібнозробити поточним. Команда cd без параметрів зробить поточним каталогомдомашній каталог.Переглянути вміст поточного або будь-якого іншого каталогу можна,скориставшись командою ls (від list). Якщо ввести її без параметрів, цякоманда роздрукує Вам список файлів, що перебувають у поточному каталозі.Якщо ж як параметр задати повне або відносне ім'я каталогуls ім'я\_директоріїто вона роздрукує список файлів у зазначеній директорії. Слід відзначити, щов отриманий список не ввійдуть файли, імена яких починаються із символу"крапка" – ".".Такі файли зазвичай створюються різними системнимипрограмами для своїх цілей (наприклад, для налаштування). Подивитисяповний список файлів можна, додатково вказавши команді ls опцію -a, тобтонабравши її у вигляді ls -a ім'я\_директорії У команди ls існує також багато інших опцій, деякі з яких ми щерозглянемо на практичних заняттях. Для одержання повної інформації прокоманду ls скористайтеся утилітою man. 1.9. Команда cat і створення файлу. Перенапрямленнявводу і виводуМи вміємо переміщуватися логічною структурою файлової системи іпереглядати її вміст. Хотілося б уміти ще й переглядати вміст файлів, істворювати їх. Для перегляду вмісту невеликого текстового файлу на екраніможна скористатися командою cat. Якщо набрати її у вигляді

cat ім'я\_файлу то на екран виведеться увесь його вміст. Не намагайтеся переглядати на екрані вміст директорій – однаковоне вийде! Не намагайтеся переглядати вміст невідомих файлів, особливоякщо Ви не знаєте, текстовий він чи бінарний. Вивід на екран бінарногофайлу може привести до непередбачуваної поведінки вашого термінала. Якщо навіть Ваш файл і текстовий, але великий, то Ви побачите тількийого останню сторінку. Великий текстовий файл зручніше переглядати задопомогою утиліти more. Якщо як параметри для команди cat задамо не одне ім'я, а іменадекількох файлівcat файл1 файл2 ... файлNто система видасть на екран їхній вміст у зазначеному порядку. Вивід командиcat можна перенаправляти з екрана термінала в який-небудь файл,скориставшись символом перенаправлення вихідного потоку даних – знаком"більше" – ">". Команда cat файл1 файл2 ... файл > файл\_результатуоб‘єднає вміст всіх файлів, чиї імена розміщені перед знаком ">", уфайл\_результату – їхню конкатинацію (від слова concatenate і походить їїназва). Прийом перенаправлення вихідних даних зі стандартного потокувиводу (екрана) у файл є стандартним для всіх команд, виконуванихкомандним інтерпретатором. Ви можете одержати файл, що містить списоквсіх файлів поточного каталогу, якщо виконаєте команду ls -a зперенаправленням вихідних даних ls -a > новий\_файлЯкщо імена вхідних файлів для команди cat не задані, то вона будевикористовувати в якості вхідних даних інформацію, що вводиться ізклавіатури, доти, поки Ви не наберете ознаку закінчення вводу – комбінаціюклавіш <CTRL> + <d>.Таким чином, команда cat > новий\_файлдозволяє створити новий текстовий файл із іменем новий\_файл і вмістимим,яке користувач введе з клавіатури. У команди cat існує багато різних опцій.Подивитися її повний опис можна в UNIX Manual.Відзначимо, що поряд з перенаправленням вихідних даних існує спосібперенаправляти вхідні дані. Якщо під час виконання деякої команди потрібноввести дані із клавіатури, можна помістити їх заздалегідь у файл, а потімперенаправляти стандартний ввід цієї команди за допомогою знака "менше" –"<" і імені файлу із вхідними даними. Інші варіанти перенаправлення потоківданих можна подивитися в UNIX Manual для командного інтерпретатора.1.10. Найпростіші команди для роботи з файлами – cp, rm,mkdir, mv

Для нормальної роботи з файлами необхідно не тільки вмітистворювати файли, переглядати їхній вміст і переміщуватися логічнимдеревом файлової системи. Потрібно вміти також створювати власніпідкаталоги, копіювати і вилучати файли, перейменовувати їх. Це мінімальнийнабір операцій, не володіючи яким не можна почувати себе впевнено прироботі з комп'ютером. Для створення нового підкаталогу використовується команда mkdir(скорочення від make directory). У найпростішому вигляді команда виглядаєтак:mkdir ім'я\_директоріїде ім'я\_директорії – повне або відносне ім'я створюваного каталогу. Укоманди mkdir є набір опцій, опис яких можна переглянути за допомогоюутиліти man.Команда cpСинтаксис командиcp файл\_джерело файл\_призначенняcp файл1 файл2 ... файлN дир\_призначенняcp -r дир\_джерело дир\_призначенняcp -r дир1 дир2 ... дирN дир\_призначенняОпис командиКоманда cp у форміcp файл\_джерело файл\_призначенняпризначена для копіювання одного файлу з іменем файл\_джерело у файліз іменем файл\_призначення.Команда cp у форміcp файл1 файл2 ... файлN дир\_призначенняпризначена для копіювання файлу або файлів з іменами файл1, файл2, ...файлN у вже існуючу директорію з іменем дир\_призначення під своїмиіменами. Замість імен файлів, що копіюються, можуть використовуватисяїхні шаблони.Команда cp у форміcp -r дир\_джерело дир\_призначенняпризначена для рекурсивного копіювання однієї директорії з іменемдир\_джерело в нову директорію з іменем дир\_призначення. Якщодиректорія дир\_призначення вже існує, то ми одержуємо команду cp унаступній форміcp -r дир1 дир2 ... дирN дир\_призначенняТака команда призначена для рекурсивного копіювання директоріїабо директорій з іменами дир1, дир2, ... дирN у вже існуючу директорію зіменем дир\_призначення під своїми власними іменами. Замість імендиректорій, які копіюються, можуть використовуватися їхні шаблони.