Міністерство освіти і науки України

Тернопільський Національний Економічний Університет

Факультет комп’ютерних інформаційних технологій

Лабораторна робота № 2

з дисципліни «Дослідження сучасних операційних систем»

Виконав: студент групи КСМм-51

Пилипчук Андрій

Перевірив: Мельник Г. М.

Тернопіль-2013

**Тема:** засоби керування процесами та мережею в ОС Linux.

**Мета**: отримати основні відомості про засоби керування процесами та мережею в ОС Linux

**Засоби для виконання роботи:**

* комп'ютер із установленою ОС Linux.

**Теоретичні відомості**

## Завантаження системи

Порядок завантаження комп’ютера (незалежно від встановленої операційної системи): після включення живлення BIOS проводить тестування апаратури, шукає завантажувальний пристрій, передає керування знайденому завантажнику, який, у свою чергу, починає завантаження операційної системи.

Після вибору операційної системи (мається на увазі Linux) завантажник передає керування ядру ОС. Ядро (kernel) починає перевірку апаратури, послідовно відображаючи на екрані інформацію про хід перевірки. Після закінчення перевірки апаратури ядро Linux може бути переведене в режим інтерактивного завантаження. Якщо не скористатися цією можливістю, то система продовжить автоматичне завантаження сервісних програм, зчитуючи інформацію з конфігураційних файлів (/etc/inittab, /etc/rc.d). Процес і результати завантаження сервісів так само відображаються на екрані. Якщо система настроєна на завантаження в графічному режимі, то в числі сервісів буде завантажений X-Сервер, інакше — система буде завантажена в текстовому режимі й на екрані з'явиться запрошення для авторизації.

У тільки що встановленій системі є всього один користувач, який може виконати вхід. Це привілейований користувач (суперкористувач) root, наділений адміністраторськими правами.

## Поняття процесу

Під процесом розуміється програма в стадії виконання. Процес — це послідовний потік, що виконується в його власному адресному просторі. Послідовний потік означає, що всередині процесу відсутня будь-яка конкуренція, тобто всі дії й операції виконуються послідовно. Виконується у власному адресному просторі, тобто пам'ять, виділена для виконання процесу, належить тільки йому й не перетинається з пам'яттю, виділеною для іншого процесу.

Процеси виконують задачі під керуванням операційної системи. Програма, як набір машинних команд і даних, збережених в виконуваному образі, на диску, являє собою пасивну (статичну) одиницю. Процес же — це програма в дії, тобто активна одиниця. Він постійно змінюється в міру виконання процесором машинних команд. Крім програмного коду й даних, процес також включає всі регістри процесора, стеки процесу, які містять тимчасові дані, параметри підпрограм, адреси повернення і т.д. Процеси є окремими задачами, кожна з яких має різні права й відповідальністю. При цьому кожний процес однозначно визначається ідентифікатором процесу, який є цілим позитивним числом.

### Стани процесу

За час свого існування процес може перебувати в різних дискретних станах. Зміна станів процесу викликається подіями. Можна виділити наступні основні стани довільного процесу.

* Стан виконання. Процес виконується, якщо в цей момент йому виділений ЦП.
* Стан готовності. Процес готовий, якщо він міг би відразу використовувати ЦП, наданий у його розпорядження.
* Стан блокування. Процес заблокований, якщо він очікує появи деякої події (наприклад, завершення операції введення-виводу), щоб одержати можливість продовжувати виконання.
* Стан припинення. У цьому стані виконання процесу припиняє, процесор йому не розподіляється, тобто процес виявляється «замороженим» у тому положенні, у якому він перейшов у стан припинення.
* Стан зомбі. Процес, що перебуває в такому стані є фактично завершеним. Однак у системі ще втримується блок керування процесом, тобто система знає про його існування.

## Керування процесами

Системи, що управляють процесами, повинні мати можливість виконувати певні операції над процесами. Основні з них містять у собі:

* створення процесу;
* завершення процесу;
* поновлення процесу;
* зміна пріоритету процесу;
* блокування процесу;
* пробудження процесу;
* запуск (вибір процесу);
* посилка сигналу процесу.

Для визначення, які процеси в системі запущені є команда ps. Якщо її запустити без усяких параметрів, то вона видає список процесів, запущених у поточній сесії. Якщо потрібно побачити список усіх процесів, запущених у системі, треба задати ту ж команду з параметром -ax.

### 1. Команда *ps*

Опції команди ps характерні тільки для GNU-версії являють собою слово, перед яким повинні стояти два дефіси. Їх не можна поєднувати, як однобуквені опції двох попередніх типів. Таким чином, існує три рівноправні формати завдання цієї команди:

ps [-опції]

ps [опції]

ps [-- довге\_ім'я\_опції [-- довге\_ім'я\_опції] ...]

Перша група опцій регулює вивід команди. Незалежно від наявності опцій цієї групи командаps видає для кожного процесу окремий рядок, але вміст цього рядка може бути різним. Залежно від заданих опцій можуть бути присутнім наступні поля:

* USER - ім'я власника процесу;
* PID - ідентифікатор процесу в системі;
* PPID - ідентифікатор батьківського процесу;
* % CPU - частина часу центрального процесора (у відсотках), виділеного даному процесу;
* % MEM - частина реальної пам'яті (у відсотках), використовувана даним процесом;
* VSZ - віртуальний розмір процесу (у кілобайтах);
* RSS - розмір резидентного набору (кількість 1 K-Сторінок у пам'яті);
* STIME - час старту процесу;
* TTY - вказівка на термінал, з якого запущений процес;
* S або STAT - статус процесу;
* PRI - пріоритет планування;
* NI - значення nice (див. опис командиniceнижче);
* TIME - скільки часу центрального процесора зайняв даний процес;
* CMD або COMMAND - командний рядок запуску програми, виконуваної даним процесом;

### 2. Команда *top*

Команда psдозволяє зробити так званий «миттєвий знімок» процесів, запущених у системі. На відміну від ps команда top відображає стан процесів і їх активність «в режимі реальному часу».

Вміст вікна обновляється кожні 5 секунд. Список процесів може бути відсортований за використовуваним часом ЦП (за замовчуванням), по використанню пам'яті, по PID, за часом виконання. Перемикати режими відображення можна за допомогою команд, які програма top сприймає. Це наступні команди (просто натискайте відповідні клавіші, тільки з урахуванням регістру, тобто разом із клавішею Shift):

* <Shift>+<N> - сортування по PID;
* <Shift>+<A> - сортувати процеси за віком;
* <Shift>+<P> - сортувати процеси по використанню ЦП;
* <Shift>+<M> - сортувати процеси по використанню пам'яті;
* <Shift>+<T> - сортування за часом виконання.

### 3. Пріоритети, значення *nice* і команда *renice*

Пріоритет для кожного процесу встановлюється в той момент, коли процес породжується. Пріоритет процесу визначається так званим "значенням nice", яке лежить у межах від +20 (найменший пріоритет, процес виконується тільки тоді, коли ніщо інше не займає процесор), до -20 (найвищий пріоритет).

Значення nice встановлюється для кожного процесу в момент породження цього процесу й при звичайному запуску команд або програм приймається рівним пріоритету батьківського процесу. Але існує спеціальна команда nice, яка дозволяє змінювати значення nice при запуску програми:

nice [- adnice] command [args]

де adnice - значення ( від –20 до +19), що додається до значення nice процесу-батька.

Отримана сума й буде значенням nice для процесу, що запускається. Негативні значення може встановлювати тільки суперкористувач. Якщо параметр -adnice не задана, то за замовчуванням для процесу-нащадка встановлюється значення nice, збільшене на 10 у порівнянні зі значенням nice батьківського процесу. Очевидно, що якщо ви не суперкористувач, то застосовувати цю команду має сенс тільки тоді, коли ви прагнете запустити якийсь процес із низьким значенням пріоритету.

Суперкористувач може змінити пріоритет будь-якого процесу в системі. Інші користувачі можуть змінювати значення пріоритету тільки для тих процесів, для яких даний користувач є власником. При цьому звичайний користувач може тільки зменшити значення пріоритету (збільшити значення nice), але не може збільшити пріоритет, навіть для повернення значення nice до значення, установлюваного за замовчуванням. Тому процеси з низьким пріоритетом не можуть породити «високопріоритетних дітей».

### 4. Сигнали й команда *kill*

Сигнали — це засіб, за допомогою якого процесам можна передати повідомлення про деякі події в системі. Самі процеси теж можуть генерувати сигнали, за допомогою яких вони передають певні повідомлення ядру й іншим процесам. За допомогою сигналів можна здійснювати такі акції керування процесами, як припинення процесу, запуск припиненого процесу, завершення роботи процесу. Усього в Linux існує 63 різних сигналу, їх перелік можна подивитися по команді

[user]$ kill –l

Сигнали прийнято позначати номерами або символічними іменами. Усі імена починаються на SIG, але цю приставку іноді опускають: наприклад, сигнал з номером 1 позначають або як SIGHUP, або просто як HUP.

Коли процес одержує сигнал, то можливий один із двох варіантів розвитку подій. Якщо для даного сигналу визначена підпрограма обробки, то викликається ця підпрограма. А якщо ні, то ядро виконує від імені процесу дія, визначене за замовчуванням для даного сигналу. Виклик підпрограми обробки називається *перехопленням* сигналу. Коли завершується виконання підпрограми обробки, процес відновляється з тієї точки, де був отриманий сигнал.

Можна змусити процес ігнорувати або блокувати деякі сигнали. Ігнорований сигнал просто відкидається процесом і не виявляє на нього ніякого впливу. Блокований сигнал ставиться в чергу на видачу, але ядро не вимагає від процесу ніяких дій до розблокування сигналу. Після розблокування сигналу програма його обробки викликається тільки один раз, навіть якщо протягом періоду блокування дане сигнал надходив кілька раз.

Команду kill суперкористувач повинен використовувати для знищення процесів-зомбі, процесів, що зависли (вони показуються в лістингу команди ps як <exiting>), процесів, які займають занадто багато процесорного часу або занадто великий обсяг пам'яті і т.д. Особливий випадок — процеси, запущені зловмисником.

**Хід роботи**

Завдання 1. Виконати всі вище перелічені команди: ps, top, renice, kill.

1. Переглянув список запущений процесів за допомогою команди ps.

2. Виконав команду top для спостереження за процесами в режимі реального часу.

3. Змінив пріоритети декількох процесів за допомогою команди renice.

4. Знищив декілька процесів командою kill.

**Висновок**

Під час виконання лабораторної роботи я отримав основні відомості про засоби керування процесами та мережею в ОС Linux, навчився користуватися командами ps, top, renice та kill.