МОНіМС України

ІФНТУНГ

Кафедра ПЗАС

**Лабораторна робота №5**

Тема: «Процеси в ОС UNIX і керування ними.»

Виконав:

cтудент групи ПІ-10-1

Пірус І.

Перевірив:

Броновський І.В.

Івано-Франківськ

2012 р.

**Тема:** Процеси в ОС UNIX і керування ними.

**Мета:** Оволодіння практичними навичками роботи з процесами — створення і знищення, керування процесами і їхній аналіз.

Створення нового процесу у системі UNIX виконується у два етапи. Спочатку системний виклик fork() викликає “розщеплення” поточного процесу на два тотожні (різниця буде лише у тому, що процес-“нащадок” має інші **PID** і **PPID**). Далі виконується системний виклик exec(), який “підміняє” контекст поточного процесу іншим контекстом. Розглянемо типовий приклад, коли один процес має запустити на виконання інший –командна оболонка sh виконує команду ls, для чого утворює новий процес, в якому виконується програма /usr/bin/ls. Для цього програміст, який пише код sh, повинен передбачити такі кроки:

1. Якщо під час синтаксичного аналізу командного рядка виявлено, що необхідно запустити на виконання деяку команду, то виконується виклик fork().
2. Наступна за fork() інструкція перевіряє значення, що повернув виклик. Якщо виклик fork() був успішний, то в результаті його виконання утворюється новий процес, тотожній батьківському, і операційна система буде виконувати їх обидва (в режимі розділення часу). Для процесу-“нащадка” значення, що повертає fork(), дорівнює 0, для батьківського процесу повертається значення PID нащадка.
3. Якщо це процес-нащадок, то виконується виклик exec(), якому в якості параметра передається ім’я файлу програми, яку необхідно запустити на виконання (в нашому прикладі – /usr/bin/ls). Якщо виклик був успішний, то на цьому виконання коду sh припиняється, і на його місце завантажується код ls.

Якщо це батьківський процес, то його дії залежать від того, які параметри були у командному рядку. Зокрема, якщо ls було запущено у пріоритетному режимі[[1]](#footnote-1), то виконання sh призупиняється до завершення процесу-нащадка, якщо ж ls було запущено у фоновому режимі, то sh видає необхідні повідомлення про запуск фонового процесу і продовжує свою роботу, тобто знову приймає і редагує командний рядок.

Процеси можуть взаємодіяти між собою за допомогою так званих сигналів. Існує обмежена кількість сигналів, які мають свої числові ідентифікатори і мнемонічні позначення. Сигнали діють як переривання, тобто вони призупиняють процес, до якого вони направлені, і викликають відповідний обробник сигналу. Деякі із сигналів мають чітко визначене значення і обробляються системним обробником. Інші можуть перехоплюватись процесом, тобто процес може встановлювати для цих сигналів свій обробник. Звичайно, існують певні узгодження щодо призначення певних сигналів, і програмістам слід дотримуватись їх при розробці своїх обробників.

Користувач також може відправити процесу сигнал, для цього існує команда **kill**. Формат команди:

**kill [-<***сигнал***>] <***PID***>**

**pkill служить для пошуку певного процесу**

pkill [-сигнал] [-fnvx] [-P список\_ppid] [-g список\_групп] [-s список\_sid] [-u список\_euid] [-U список\_uid] [-G список\_групп] [-J список\_projid] [-t список\_терминалов] [-T список\_taskid] [шаблон]

**fg** команда для виклику процеса із фонового режиму.

**bg** повертає команду назад у фоновий режим.

**jobs** - Перераховує ваші завдання.

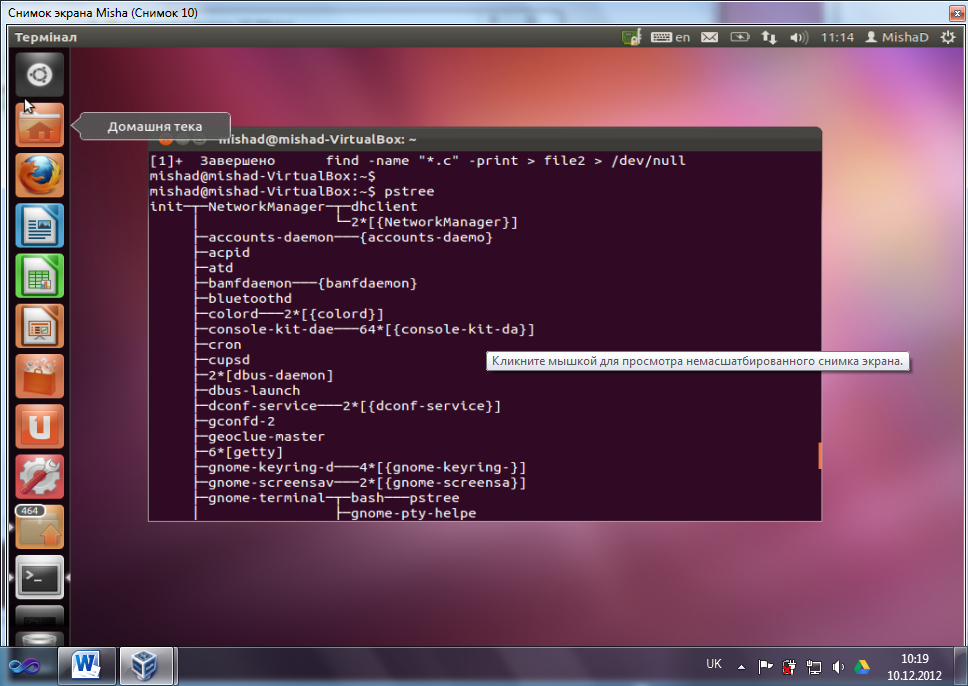
**&** - Виконати завдання у фоновому режимі.

Виконання:

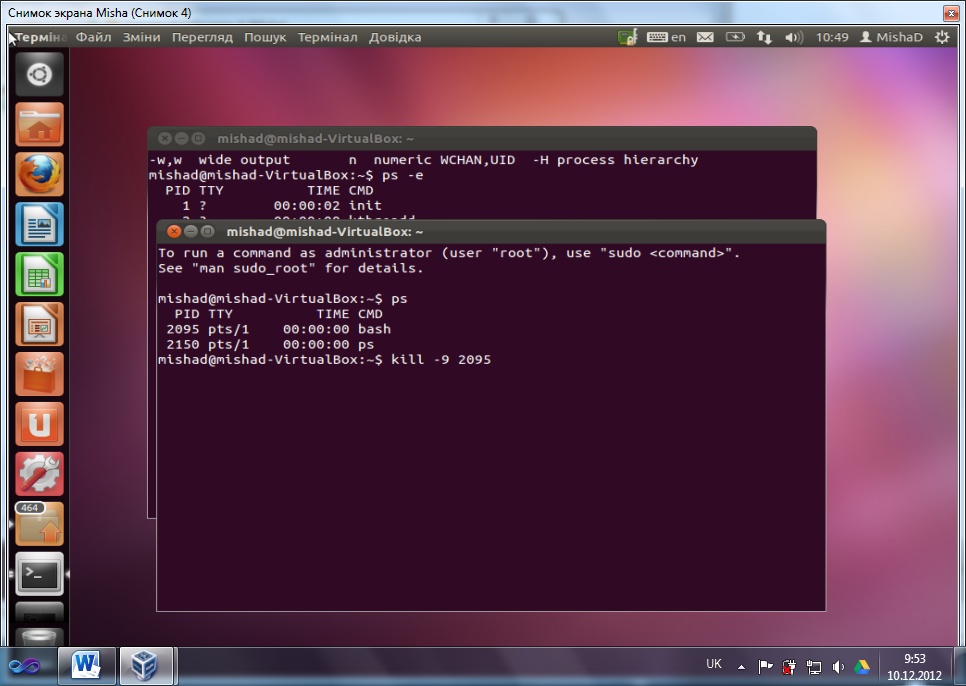
1.Переглядаю розширений список запущених мною процесів використовуючи команду ps –U mishad –u misha u.

2.Переглядаю повний список запущених процесів у системі за допомогою команди ps -ALF.

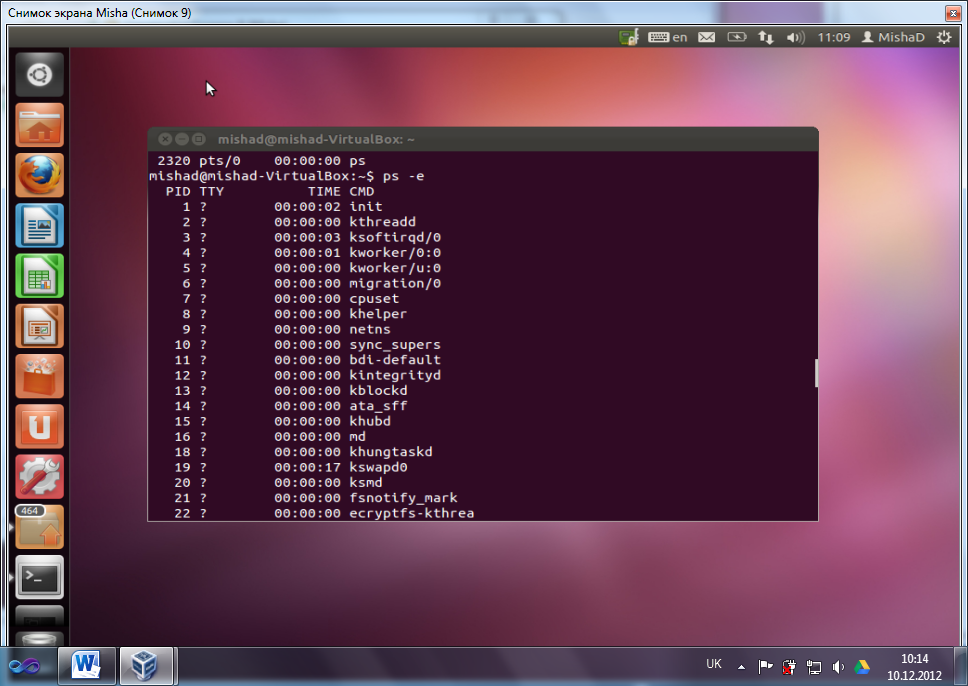
За допомогою команди pstree переглядаю ієрархію процесів.



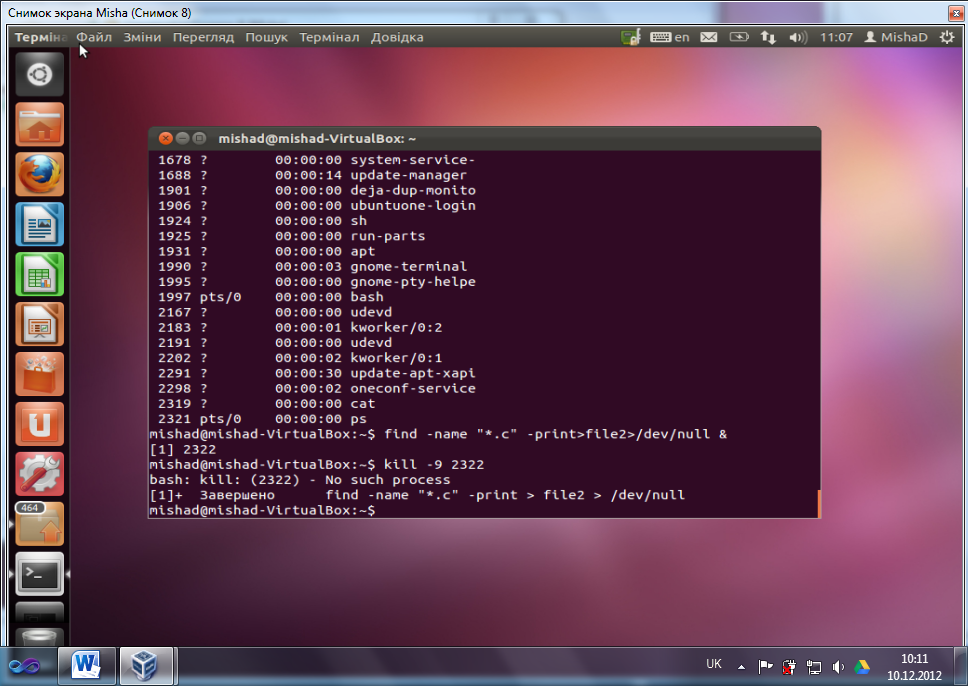
3.Запускаю ще один shell, переглядаю запущені процеси визначаю який PID відповідає процесу shell і за допомогою команди kill знищую цей процес.



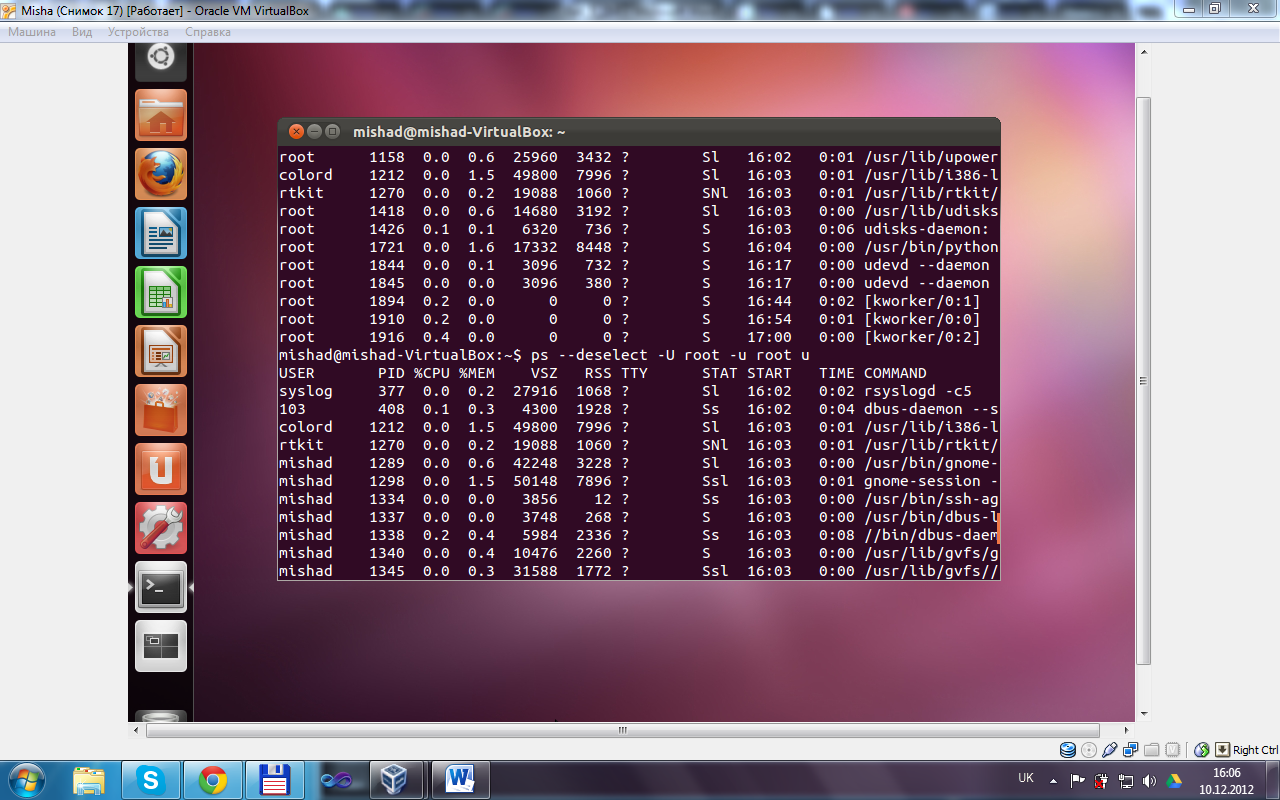
4.Переглядаю весь список задач за допомогою команди ps –e.



5.Запускаю фоновий процес , після чого за допомогою команди kill я його завершую.

6

6. За допомогою команди ps –deselect –U root –u root u переглядаю список запущених процесів не користувачем root.



**Висновок:** на даній лабораторній роботі я навчився працювати з процесами та керувати ними.

1. [↑](#footnote-ref-1)