Міністерство освіти і науки, молоді та спорту України

Івано-Франківський Національний Технічний Університет Нафти і Газу

Кафедра ПЗАС

**Лабораторна робота №5**

**Процеси в ОС UNIX та керування ними**

Виконала:

ст. групи ПІ-10-1

Креховецька І.В.,

Перевірив:

Броновський І.В.

Івано-Франківськ

2012

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

**Завдання:**

1. Перегляньте список процесів користувача (Вас).
2. Перегляньте повний список процесів, запущених у системі. При цьому гарантуйте збереження інформації від "утікання" з екрана (якщо процесів багато). Зверніть увагу на ієрархію процесів. Простежте через поля **PID** і **PPID** всю ієрархію процесів тільки-но запущеної Вами команди, починаючи з початкового процесу **init**. Зверніть увагу на формування інших полів виводу.
3. Запустіть ще один **shell**. Перегляньте повний список процесів, запущених вами, при цьому зверніть увагу на ієрархію процесів і на їхній зв'язок з терміналом. Використовуючи команду **kill**, завершіть роботу в цьому **shell**’і.
4. Перегляньте список задач у системі і проаналізуйте їхній стан.
5. Запустіть фоновий процес командою

**find / -name "\*.c" -print > file 2> /dev/null &**

1. Визначте його номер. Відправте сигнал призупинення процесу. Перегляньте список задач у системі і проаналізуйте їхній стан. Продовжить виконання процесу. Знову перегляньте список задач у системі і проаналізуйте його зміну. Переведіть процес в активний режим, а потім знову у фоновий. Запустіть цей процес із пріоритетом 5.
2. Виведіть на екран список усіх процесів, запущених не користувачем ***root***.
3. Організуйте вивід на екран календаря <1996+№варіанту> року через 1 хвилину після поточного моменту часу.
4. Організуйте періодичне (щоденне) видалення в домашньому каталозі усіх файлів з розширенням **\*.bak** і **\*.tmp**.

## Короткі теоретичні відомості:

UNIX – багатозадачна система з розділенням часу. Це означає, що в системі одночасно виконується багато процесів. Кожний процес асоційований з певним користувачем, від імені якого цей процес діє. Для того, щоби переглянути список процесів, існує команда **ps**. Ця команда має багато ключів-модифікаторів, які визначають, яку саме інформацію про процеси повинна виводити команда. Слід зазначити, що в різних системах UNIX значення цих ключів може суттєво відрізнятись. Типові ключі: **-a** виводить інформацію про всі процеси, а не лише про процеси даного користувача, **-l** та **-x** виводять розширену інформацію про процес.

Кожний процес в системі має свій унікальний ідентифікатор – **PID**. За цим ідентифікатором можна звертатись до процесу. Крім того, кожний процес виникає не сам по собі – він має так званий батьківський процес, що характеризується ідентифікатором **PPID** (*parent process ID*). Таким чином утворюється ієрархія процесів, що бере початок від початкового процесу **init**.

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***>**

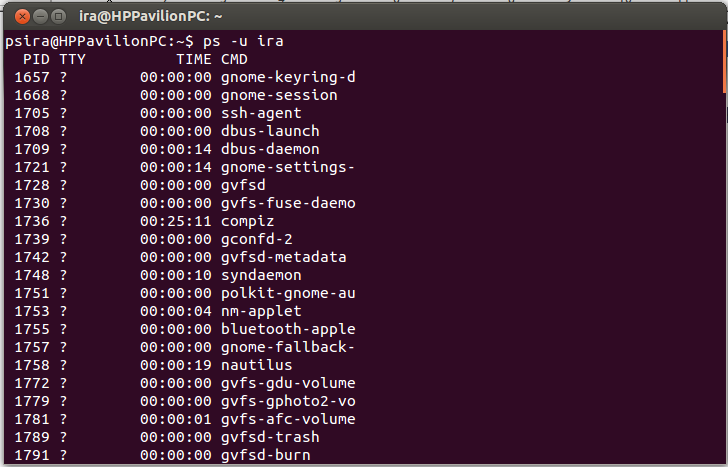
**<***сигнал***>** – це мнемонічне або числове позначення сигналу (наприклад, **STOP**, **TERM**, **CONT**, **9**), а **<***PID***>** –ідентифікатор процесу, якому посилають сигнал. Якщо не вказати параметр **<***сигнал***>**, то буде відправлено сигнал завершення процесу **TERM** (**15**). Цей сигнал може перехоплюватись процесом, але існує сигнал **KILL** (**9**), який не перехоплюється і безумовно знищує процес (якщо у користувача є достатньо для цього прав). Таким чином можна зупинити будь-який свій процес, якщо над ним втрачене керування (або процес “завис”, що у системі UNIX трапляється дуже нечасто, або користувач не знає, яку команду процес може сприйняти). Для цього слід зайти з іншої консолі (віртуального або фізичного терміналу) і дати команду **kill -9 <***PID***>**, де **<***PID***>** можна дізнатись за допомогою попередньої команди **ps**.

Коли користувач дає команду з консолі, в системі запускається процес, або кілька процесів. Якщо в командному рядку утворюється конвеєр (наприклад, **ls –l | wc –l**), то всі процеси (у нашому прикладі **ls** і **wc**) запускаються одночасно. Разом вони складають так зване завдання. Завдання пов’язано з терміналом. Поки воно не завершиться, користувач не має можливості вводити наступну команду. Це так званий пріоритетний (*foreground*) режим виконання завдання. Щоби під час виконання завдання мати можливість запускати інше завдання, перше з них слід запустити в так званому фоновому (*background*) режимі. Для того, щоби запустити завдання в фоновому режимі, слід завершити рядок команди символом **“&”** (після пробілу). При цьому стандартний ввід за замовчанням назначається порожньому файлу **/dev/null**. Слід врахувати, що завдання в фоновому режимі може намагатись здійснювати вивід на екран, заважаючи при цьому виводу пріоритетного процесу (спробуйте працювати, запустивши у фоновому режимі команду **ping**). Тому слід подбати, щоби фонові завдання здійснювали вивід у файли (див. Лабораторну роботу №4). Завдання, що було запущено у пріоритетному режимі, можна перевести у фоновий. Для цього необхідно спочатку призупинити виконання завдання (комбінація клавіш **CTRL-Z**). Далі можна поновити виконання завдання у пріоритетному режимі (команда **fg**) або у фоновому режимі (команда **bg**). Завдання мають свої номери. Переглянути їх можна за допомогою команди **jobs**.

**Виконання:**

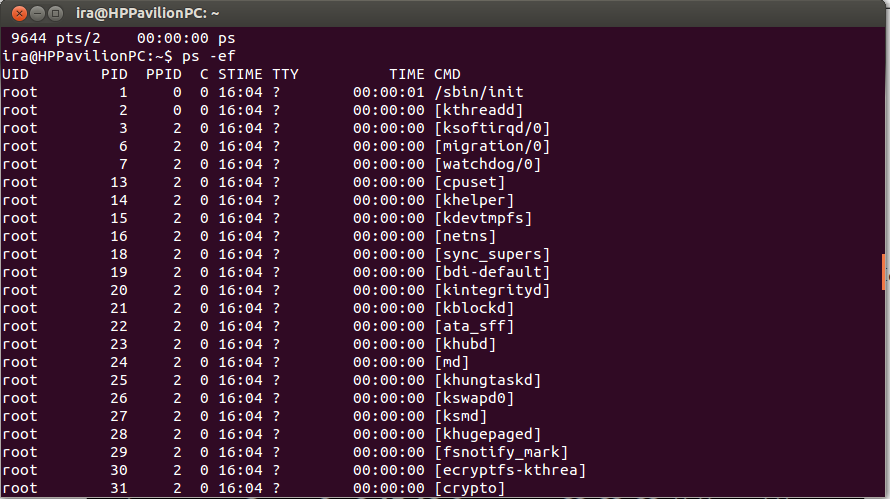
1. Переглянемо список процесів користувача ira. Для цього виконаємо команду

ps –u ira

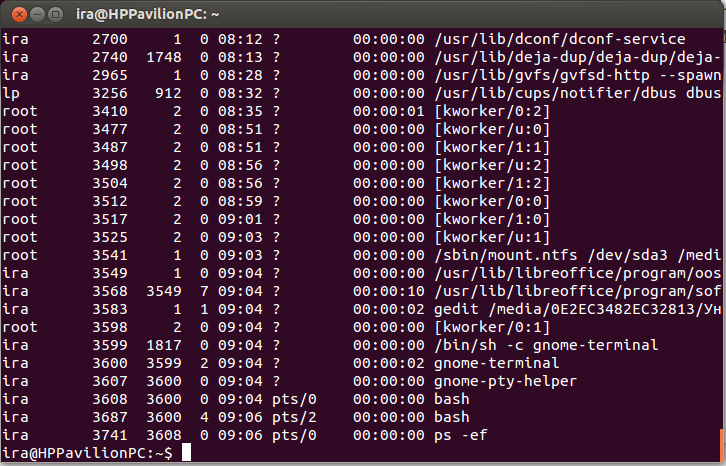


1. Переглянемо повний список процесів. Команда:

ps -ef



1. Запустимо ще один shell та виконаємо попередній пункт ще раз. Знайдемо в списку команд що виконуєються запущений shell.



Як видно, PID запущеного shell-у 3687. Щоб закрити його виконаємо команду

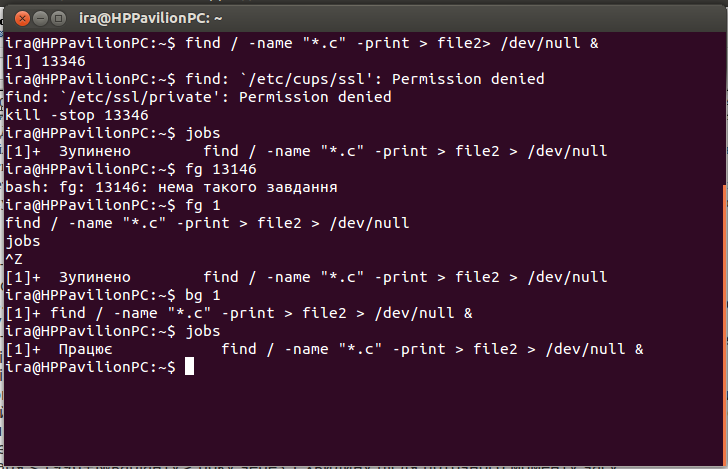
kill -kill 3687

1. Переглянемо список задач у системі за допомогою комади jobs.
2. Запустимо фоновий процес командою

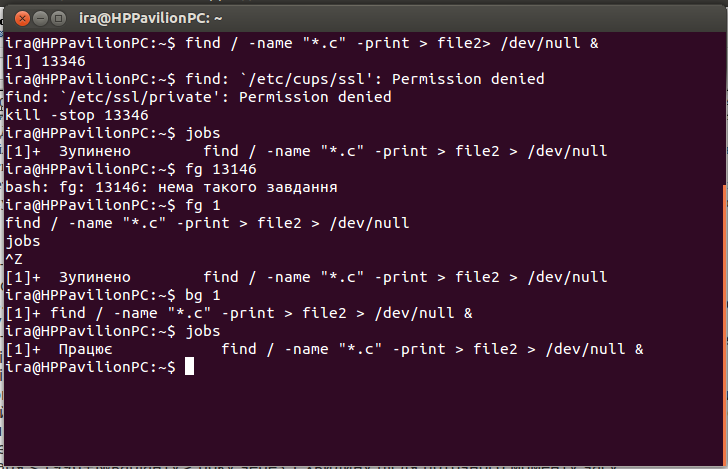
find / -name "\*.c" -print > file 2> /dev/null &

Команда почала виконувати у фоновому режимі.

1. Відправимо сигнал призупинення процесу командою kill -stop 13346. Після цього переглянемо список працюючих поцесів за допомогою команди jobs та переконаємось що потрібний нам процес зупинений.



Після цього запустимо процес у пріорітетному режимі. Для цього виконаємо команду fg 1. Зупинемо виконнаня цього процесу у пріорітетному режимі командою Ctrl-Z. Та знову запустимо, але у фоновому режимі командою bg 1.



1. Виведемо на екран список всіх процесів запущених не користувачем root.

ps -ef -u root

1. Організуємо вивід на екран календаря 2006 року. Для цього створимо скрипт з наступним текстом:

#!bin/bash #!/bin/bash

echo "Calendar $(cal 9 1998)”

Та виконаємо команду:

at –f cal.sh –v 21:25

1. Організуємо поденне видалення в домашньому каталозі усіх файлів з розширенням \*.bak і \*.tmp.

#!/bin/sh

find /home/ -name "\*.bak" -delete

find /home/ -name "\*.tmp" –delete

Перемістимо його в /usr/local/bin/:

sudo mv ~/delscr /usr/local/bin/ delscr

Робимо його виконуваним:

sudo chmod +x /usr/local/bin/delscr

Робимо запис в crontab

crontab -e

0 6 \* \* \* sh /usr/local/bin/delscr

**Висновок:** Я навчилась працювати з процесами в ОС Linux.