МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Одеський національний політехнічний університет

Інститут комп'ютерних систем

Кафедра інформаційних наук

Звіт

Лабораторна робота №8

з дисципліни «Операційні системи»

Тема: «Програмування керуванням процесами в ОС Unix»

Виконал

Студент АІ-204

Черемних Г.В.

Перевірили

Одеса 2021

Мета роботи: отримання навичок в управлінні процесами в ОС Unix на рівні мови програмування С.

Завдання до виконання

Варіант 3

Завдання 1 Перегляд інформації про процес

Створіть C-програму, яка виводить на екран таку інформацію:

− ідентифікатор групи процесів лідера сесії;

− ідентифікатор групи процесів, до якої належить процес; − ідентифікатор процесу, що викликав цю функцію; − ідентифікатор батьківського процесу;

− ідентифікатор користувача процесу, який викликав цю функцію;

− ідентифікатор групи користувача процесу, який викликав цю функцію.

При створенні повідомлень використовуйте функцію fprintf з виведенням цих повідомлень на потік помилок. Після компіляції запустіть програму та перегляньте результат її роботи. Запустіть два рази тут ж саму програму в конвеєрі, наприклад: ./info | ./info Перегляньте результат її роботи та порівняйте значення ідентифікатору групи процесів для різних процесів. Запустіть два рази тут ж саму програму, але у послідовності без конвеєру, наприклад: ./info ; ./info Перегляньте результат її роботи та порівняйте значення ідентифікатору групи процесів для різних процесів.

Завдання 2 Стандартне створення процесу Створіть C-програму, яка створює процес-нащадок, породжуючи процес та замінюючи образ процесу.

У програмі:

1) процес-батько повинен видати повідомлення типу «Parent of Ivanov», використовуючи стандартний потік виводу, або помилок;

2) процес-нащадок повинен видати повідомлення типу «Child of Ivanov», але вже через виклик команди echo з функції execve, де замість слова Ivanov в повідомленні повинно бути ваше прізвище в транслітерації

Завдання 3 Обмін сигналами між процессами

3.1 Створіть C-програму, в якій процес очікує отримання сигналу SIGUSR2 та виводить повідомлення типу «Process of Ivanov got signal» після отримання сигналу, де замість слова Ivanov в повідомленні повинно бути ваше прізвище в транслітерації. Запустіть створену С-програму.

3.2 Створіть C-програму, яка надсилає сигнал SIGUSR2 процесу, запущеному в попередньому пункту завдання. Запустіть створену С-програму та проаналізуйте повідомлення, які виводить перша програма. Завершіть процес, запущений в попередньому пункті завдання.

Завдання 4 Створення процесу-сироти Створіть C-програму, в якій процес-батько несподівано завершується раніше процесу-нащадку. Процес-батько повинен очікувати завершення n+1 секунд. Процеснащадок повинен в циклі (2\*n+1) раз із затримкою в 1 секунду виводити повідомлення, наприклад, «Parent of Ivanov», за шаблоном як в попередньому завданні, і додатково виводити PPID процесу-батька. Значення n – номер команди студента + номер студента в команді. Перевірте роботу програми, вивчіть вміст таблиці процесів і зробіть відповідні висновки.

Завдання 5 Створення процесу-зомбі Створіть C-програму, в якій процес-нащадок несподівано завершується раніше процесу-батька, перетворюється на зомбі, виводячи в результаті повідомлення, наприклад, «I am Zombie-process of Ivanov», за шаблоном як в попередньому завданні. Запустіть програму у фоновому режимі, а в окремому терміналі вивчіть вміст таблиці процесів і зробіть відповідні висновки.

Завдання 6 Попередження створення процесу-зомбі Створіть C-програму, в якій процес-нащадок завершується раніше процесу-батька, але ця подія контролюється процесом-батьком. Процес-нащадок повинен виводити повідомлення, наприклад, «Child of Ivanov is finished», за шаблоном як в попередньому завданні. Процес-батько повинен очікувати (3\*n) секунд. Значення n - n – номер команди студента + номер студента в команді. Запустіть програму у фоновому режимі, а в окремому терміналі вивчіть вміст таблиці процесів і зробіть відповідні висновки.

3. Виконання програми

**Висновок** : Під час виконання лабораторної роботи я придбав навичок в управлінні процесами в ОС Unix на рівні мови програмування С.