

Systemy operacyjne 2016

Lista zadań nr 4

Na zajęcia ?

Studenci są zachęceni do przeprowadzania dodatkowych eksperymentów związanych z treścią zadań i dzieleniem się obserwacjami z resztą grupy. Proszę najpierw korzystać z podręcznika systemowego (polecenia `man` i `apropos`) i w razie potrzeby sięgać do zasobów Internetu. Głównym podręcznikiem do zajęć praktycznych jest „*The Linux Programming Interface: A Linux and UNIX System Programming Handbook*”. Należy zapoznać się z treścią rozdziału 2 w celach poglądowych, a resztę książki czytać w razie potrzeby. Bardziej wnikliwe wyjaśnienia zagadnień można odnaleźć w książce „*Advanced Programming in the UNIX Environment*”.

Rozwiązania należy starannie przygotować w domu – najlepiej gdyby były w postaci pliku tekstowego z listą poleceń do wykonania i komentarzami. Do prezentacji każdego zadania należy utworzyć osobą zakładkę w terminalu. Na początku zajęć student deklaruje przygotowane zadania. Prowadzący podchodzi do kolejnych studentów i odpytuje z jednego wybranego zadania. Należy być przygotowanym do wyjaśnienia pojęć oznaczonych **wytluszczoną** czcionką. W przypadku zbędnego przeciągania czasu odpowiedzi lub niedostatecznego przygotowania student może nie otrzymać punktów za zadanie. Procedura będzie kontynuowana aż do wyczerpania czasu przeznaczonego na zajęcia.

Zadanie 1. Sprawdź i szczegółowo opisz działanie polecenia `ps` z opcjami: `-A`, `-a`, `-uuser`, `-Ccmd`. Jak z jego pomocą znaleźć **identyfikator procesu**, **grupy**, **rodzica** albo **terminal sterujący**? Kto jest rodzicem procesu `init`? Wskaż, które z wyświetlonych zadań są wątkami jądra. Jakie jest znaczenie poszczególnych znaków w kolumnie `STAT`? Wyświetl drzewiastą strukturę procesów poleceniem `pstree` – które ze wskazanych zadań są wątkami?

Zadanie 2. Uruchom wybrany proces, a następnie zakończ go poleceniem `kill`. Który sygnał wysła polecenie `kill` bez podania dodatkowych argumentów? Zaprezentuj inne metody wysyłania sygnałów, tj. `xkill` i `pkill`. Procesy mogą ignorować niektóre sygnały – jak zatem wymusić ich zakończenie? Jak przy pomocy polecenia `pgrep` zliczyć ilość procesów danego użytkownika?

Zadanie 3. Zaprezentuj sytuację, w której proces zostanie **osierocony**. W terminalu uruchom dodatkową kopię powłoki `bash`. Z jej poziomu wystartuj w tle¹ program `xclock` i sprawdź, kto jest jego rodzicem. Poleceniem `kill` wyślij sygnał `SIGKILL` do uruchomionej wcześniej powłoki i sprawdź, kto stał się nowym rodzicem procesu `xclock`. Wyjaśnij przebieg i wyniki powyższego eksperymentu. Co się stanie, gdy zamiast `SIGKILL` wyślemy powłóce sygnał `SIGHUP`?

Zadanie 4. Czym jest **grupa procesów**, **grupa procesów pierwszoplanowych**, **sesja** i **terminal**? Pokaż wszystkie zadania należące do **bieżącej sesji**, a potem **przyłączone** do bieżącego terminala. Jaki jest związek między **identyfikatorem sesji**, a **terminalem sterującym**? Wyświetl listę wszystkich zadań i zidentyfikuj procesy będące **przywódcami sesji**. Znajdź procesy działające w obrębie jednej sesji i jednej grupy procesów. Czemu wprowadzono to rozróżnienie?

Zadanie 5. Wykonaj wybrany program i **poleceniem powłoki** `echo $?` zbadaj jego **kod wyjścia**. Jaką wartość przyjmie kod wyjścia jeśli program: wykonał się poprawnie albo nie, został zakończony przy pomocy sygnału. Jak zinterpretować kod wyjścia w drugim przypadku?

¹Dodając znak `&` na końcu linii poleceń.

Zadanie 6. Znajdź identyfikator PID jednego ze swoich procesów, po czym wydrukuj zawartość katalogu `/proc/${PID}`². Wyświetl plik zawierający **argumenty wywołania, zmienne środowiskowe** i opis przestrzeni adresowej badanego procesu (`maps`). Wyjaśnij znaczenie kolumn (`man proc`) i wskaż, gdzie znajduje się `stext`, `stos`, `segment text`, `data` oraz `bss` oraz procedury dynamicznego konsolidatora. Jaka jest rola segmentu `VDSO`?

Zadanie 7. Uruchom program `xeyes`, a następnie przy pomocy odpowiedniej kombinacji klawiszy wyślij mu sygnał `SIGSTOP`. W jaki sposób wznowić wykonanie programu? Przeprowadź inspekcję pliku `/proc/${PID}/status` i wyświetl **maskę sygnałów** zgłoszonych procesowi (ang. *pending signals*). Pokaż jak będzie się zmieniać, gdy będziemy wysyłać wstrzymanemu procesowi kolejne sygnały, tj. `SIGUSR1`, `SIGUSR2`, `SIGHUP`, `SIGINT`. Co opisują pozostałe pola pliku `status` dotyczące sygnałów? Który sygnał zostanie dostarczony jako pierwszy po wybudzeniu procesu?

Zadanie 8. Uruchom aplikację `firefox` i przy pomocy programu `lsuf` wyświetl zasoby należące do procesu przeglądarki. Podaj znaczenie poszczególnych kolumn wykazu i zidentyfikuj, które z wymienionych zasobów są **zwykłymi plikami, katalogami, urządzeniami, gniazdami sieciowymi, gniazdami domeny uniksowej, potokami**. Przechwyć wyjście z programu `lsuf` przed i po otwarciu wybranej strony w nowej zakładce, po czym wyświetl różnice poleceniem `diff -u`.

Zadanie 9. Zapoznaj się z poleceniami `strace` i `ltrace`. Uruchom wybrany program w trybie śledzenia wywołań systemowych i wywołań bibliotecznych³. Podłącz się do wybranego procesu i obserwuj jego działanie. Jak śledzić aplikacje złożone z wielu procesów lub wątków? Jak zliczyć ilość wywołań systemowych, które wykonał proces w trakcie swego wykonania? Jak obserwować wyłącznie pewien podzbiór wywołań systemowych, np. `open`, `read` i `write`?

Zadanie 10. Przy pomocy **wbudowanego polecenia powłoki** `time` zmierz czas wykonania wybranego procesu, np. polecenia `find /var`. Cemu czasy `user` i `sys` nie sumują się do `real`? Przy użyciu wbudowanego polecenia powłoki `ulimit` nałóż ograniczenie **czasu wykonania** na procesy potomne. Który sygnał jest wysyłany do procesu po przekroczeniu ograniczeń? Pamiętaj, że po nałożeniu limitów nie można ich cofnąć w obrębie tej samej instancji powłoki.

Zadanie 11. Wyświetl wykaz uruchomionych procesów wraz z ich **priorytetem** oraz wartością `nice`. Poleceniem `renice` spróbujemy wpłynąć na decyzje planisty zadań. Celem wygenerowania sztucznego obciążenia procesora uruchom polecenie `echo "" | awk '{for(;;) {}}'` w kilku terminalach jednocześnie. Wystartuj przeglądarkę `firefox` i z użyciem `renice` zmniejsz jej priorytet. Jak zmienił się **czas reakcji procesu**? Czy możesz przywrócić wartość `nice` do poprzedniego stanu?

²Notacja zmiennych powłoki. Wystąpienie ciągu znaków `${symbol}` jest zamieniane na wartość zmiennej `symbol`.

³Konfiguracja systemu może wymagać użycia polecenia `sudo` do uruchomienia programów śledzących.