

Systemy operacyjne

Warsztaty 1

Ćwiczenie 1

Otwórz dokumentację polecenia `ps` i pobieżnie zapoznaj się z nią. Pokaż zadania należące do bieżącej sesji, użytkownika oraz wszystkie procesy. Wskaż, które z wyświetlonych zadań są wątkami jądra. Wyświetl drzewiastą strukturę procesów, a następnie do listingu dodaj wątki. Przyjrzyj się procesowi `init`, kto jest jego rodzicem? Czy można go zakończyć?

Ćwiczenie 2

Znajdź identyfikator `pid` jednego ze swoich procesów. Następnie wylistuj katalog `/proc/$pid`. Zapoznaj się z dokumentacją wirtualnego systemu plików `proc`. Wyświetl plik `maps` i przeanalizuj jego zawartość. Wskaż gdzie znajduje się `sterta`, `stos`, `segment text / data / bss` programu oraz pozostałości dynamicznego konsolidatora. Jakie inne użyteczne informacje można odnaleźć w katalogu procesu?

Ćwiczenie 3

Uruchom jakiś proces, a następnie zakończ go poleceniem `kill`. Znajdź wygodniejszą metodę kończenia procesów (np. `xkill`, `pkill`). Zapoznaj się z dokumentacją polecenia `pgrep`, służącego do filtrowania procesów zgodnie z pewnymi kryteriami. Jaki sygnał wysyła domyślnie polecenie `kill`? Niektóre sygnały proces może przechwycić. Jak skutecznie zakończyć dany proces? Znajdź numer sygnału służący do przeładowywania konfiguracji demonów.

Ćwiczenie 4

Prześledź listę rodziców (przodków) bieżącego procesu. Co się stanie jeśli proces utraci swojego rodzica? Czym jest identyfikator sesji i jaki ma związek z terminalem kontrolującym? Wyświetl listę wszystkich zadań i zidentyfikuj procesy będące przywódcami sesji. Znajdź procesy działające w obrębie jednej sesji i jednej grupy procesów. Czemu wprowadzono to rozgraniczenie?

Ćwiczenie 5

Wykonaj jakieś polecenie i zbadaj jego kod wyjścia. Co się stanie, jeśli zakończysz proces przy pomocy sygnału? Jakie warunki muszą zajść, by proces stał się zombie? Użyj polecenia `ps` by wyświetlić wszystkie procesy, ich stan, oraz w jakim fragmencie jądra oczekują na wznowienie.

Ćwiczenie 6

Większość zasobów w systemach uniksowych ma semantykę pliku. Zapoznaj się z dokumentacją programu `lsdf`. Otwórz przeglądarkę internetową i wyświetl wszystkie otwarte pliki należące do tego procesu. Zidentyfikuj, które z zasobów są plikami, katalogami, urządzeniami, gniazdkami sieciowymi. Przeanalizuj właściwości każdego z plików.

Ćwiczenie 7

Zapoznaj się z dokumentacją do poleceń `strace` i `ltrace`. Uruchom jakiś prosty program w trybie śledzenia wywołań systemowych i wywołań bibliotecznych. Podłącz się do jakiegoś działającego procesu i obserwuj jego działanie. Jak śledzić aplikacje złożone z wielu procesów lub wątków? Jakiego wywołania systemowe jest używane do implementacji śledzenia?

Ćwiczenie 8

Znajdź dwa programy - jeden skonsolidowany statycznie, a drugi dynamicznie. Odczytaj strukturę plików wykonywalnych tych programów przy pomocy polecenia `readelf` lub `objdump`. Jakich bibliotek wymaga drugi program, aby zostać załadowanym? Pomocne będzie narzędzie `ldd`. Zauważ, że `ldd` informuje również pod jakie adresy byłyby ładowane biblioteki. Czy adresy te zmieniają się z każdym wywołaniem `ldd`?