

Systemy operacyjne

Ćwiczenia 3

Zadanie 1

Opisz przestrzeń adresową procesu w systemach uniksowych. Zdefiniuj pojęcie obrazu procesu. Czemu z reguły pierwsze kilka megabajtów przestrzeni użytkownika jest niedostępne? Jakie niebezpieczeństwa niesie za sobą umieszczanie kodu i danych zawsze pod takimi samymi adresami? Jaki problem sprawia ładowanie bibliotek dynamicznych do przestrzeni adresowej procesu?

Zadanie 2

Opisz zasoby przechowywane w bloku kontrolnym procesu (ang. PCB - process control block). Uzasadnij czemu te informacje muszą być powiązane z procesem. Podaj argumenty za tym, że część informacji skojarzonych z procesem można przechowywać bezpiecznie w przestrzeni użytkownika. Jakie korzyści to przynosi?

Zadanie 3

Opisz pięciostanowy model procesu. Wyjaśnij, które przejścia w automacie opisującym ten model są inicjowane przez system operacyjny, a które przez proces. Podaj jakie akcje wymuszają zmianę stanów? Które z decyzji po stronie systemu operacyjnego będą podejmowane przez planistę krótkoterminowego i długoterminowego.

Zadanie 4

Rozszerz model pięciostanowy, tak by umożliwić operacje wymiany pamięci procesu na dysk (ang. swapping). Dlaczego system operacyjny mógłby chcieć oddelegować pamięć procesu na dysk? W kontekście nowych stanów, odpowiedz na pytania z zadania 3.

Zadanie 5

Na przykładzie systemów uniksowych opisz dokładnie proces uruchomienia programu z dysku. Upewnij się, że nie pomijasz żadnego ważnego etapu (np. tworzenie przestrzeni adresowej, przypisywanie zasobów, działanie dynamicznego konsolidatora). Czym różni się tworzenie procesów w systemie Linux i Windows NT? Rozważ wady i zalety obu rozwiązań.

Zadanie 6

Zdefiniuj przełączanie procesów i opisz jak zachodzi taki proces. Wyjaśnij jakie są przyczyny przerywania wykonania procesów i do czego mogą służyć. Zklasyfikuj i podaj przykłady zdarzeń, z powodu których system operacyjny może zakończyć proces. Czy są takie zdarzenia, które normalnie uznaje się za błąd programu, ale potencjalnie mogą być obsłużone przez aplikację lub system operacyjny celem implementacji pewnych funkcji? Jeśli tak, to podaj przykład.

Zadanie 7

Wymień trzy modele wykonania kodu jądra. Opisz po krótko różnice między nimi - podaj wady i zalety każdego rozwiązania (skup się na wydajności, bezpieczeństwie i sposobie komunikacji użytkownik-jądro). Dla każdego modelu podaj co najmniej jeden system operacyjny, który go implementuje.

Zadanie 8

Opisz zmiany w modelu stanowym wątków względem pięciostanowego modelu procesów. Czemu przełączanie wątków jest operacją szybszą niż przełączanie procesów? Rozważ poprzednie pytanie w kontekście systemów wieloprocesorowych. Zdefiniuj pojęcie powinowactwa wątków (ang. affinity) i podaj przykład jak taki mechanizm może wspomóc pisanie aplikacji wielowątkowych.

Zadanie 9

Opisz różnice między wątkami przestrzeni jądra (ang. KLT - kernel level thread), a wątkami przestrzeni użytkownika (ang. ULT - user level thread). Podaj zalety KLT. Jakie korzyści lub problemy niesie ze sobą użycie ULT. Jak biblioteka wątków ULT musi skompensować brak wsparcia jądra? Zdefiniuj pojęcie obwolutowania (ang. jacketing). Rozważ model hybrydowy i pokaż, że może on połączyć zalety obu rozwiązań (tj. KLT i ULT).

Zadanie 10

Rozważmy wątki przestrzeni jądra. Które z zasobów należących do procesu są współdzielone przez wątki, a które są przydzielone osobno dla każdego wątku? Uzasadnij swoją odpowiedź. Znajdź informacje o lokalnej pamięci wątku (ang. TLS - thread local storage) i opisz ten mechanizm.