Systemy operacyjne

Ćwiczenia 3

Terminy: przestrzeń adresowa, mapa pamięci (ang. *memory layout / map*), obraz procesu (ang. *process image*), segment kodu / danych (ang. *text / data segment*), tablice procesów, blok kontrolny procesu (ang. *PCB - process control block*), planista krótkoterminowy / długoterminowy (ang. *short-term / long-term scheduler*), stan procesu (ang. *process state*), wymiana (ang. swapping), sygnały uniksowe, potomek (ang. *child process*), rodzic (ang. *parent process*), tworzenie procesu (ang. *process spawning*), wielowątkowość (ang. *multithreading*), przetwarzanie symetryczne (ang. *symmeric multiprocessor*), powinowactwo (ang. *affinity*), wątki przestrzeni jądra (ang. *KLT - kernel level thread*), wątki przestrzeni użytkownika (ang. *ULT - user level thread*), obwolutowanie (ang. *jacketing*), pamięć lokalna wątku (ang. *TLS - thread local storage*).

Zadanie 1

Opisz szczegółowo mapę pamięci procesu w systemie Linux. Zdefiniuj pojęcie obrazu procesu. Czemu z reguły pierwsze kilka megabajtów przestrzeni użytkownika jest niedostępne dla programu? Jakie niebezpieczeństwa niesie ze sobą umieszczanie kodu i danych zawsze pod takimi samymi adresami w przestrzeni procesu? Jaki problem sprawia ładowanie bibliotek dynamicznych do przestrzeni adresowej procesu?

Zadanie 2

Opisz zasoby przechowywane w bloku kontrolnym procesu. 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 rezultatem działań podejmowanych przez (a) system operacyjny (b) proces użytkownika. 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. Dlaczego system operacyjny mógłby chcieć odesłać 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 ono przebiega w systemach uniksowych. Wyjaśnij jakie są przyczyny przerywania wykonania procesów i czemu mogą służyć. Sklasyfikuj typy 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

Rozważmy wątki przestrzeni jądra w systemie Linux. Czemu przełączanie wątków jest operacją szybszą niż przełączanie procesów? Rozważ poprzednie pytanie w kontekście systemów wieloprocesorowych. Zdefiniuj pojęcie powinowactwa wątków i podaj przykład jak taki mechanizm może wspomóc pisanie aplikacji wielowątkowych. Znajdź informacje o lokalnej pamięci wątku i opisz ten mechanizm.

Zadanie 8

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