Systemy operacyjne

wykład 6 - architektura systemu komputerowego a struktura systemu operacyjnego

dr Marcin Ziółkowski

Instytut Matematyki i Informatyki
Akademia im. Jana Długosza w Częstochowie

21 kwietnia 2016 r.

ELEMENTY ARCHITEKTURY SYSTEMU KOMPUTEROWEGO

Współczesny komputer składa się przede wszystkim z następujących elementów:

- procesor
- urządzenia zewnętrzne
- sterowniki urządzeń zewnętrznych
- pamięć operacyjna
- pamięć podręczna
- sterownik pamięci
- magistrala systemowa

ROLA STEROWNIKÓW I MAGISTRALI

STEROWNIK odpowiada za nadzorowanie pracy określonej grupy urządzeń, pośredniczy w przesyłaniu informacji do urządzeń oraz w kierunku przeciwnym. Posiada swoją własną pamięć, co pozwala na szybkie przesyłanie informacji magistralą systemową niezależnie od szybkości danego urządzenia.
W systemie obecnych jest często wiele MAGISTRAL niższego poziomu, które są połączone z MAGISTRALĄ systemową poprzez sterowniki. Urządzenia również mogą być połączone ze swoimi sterownikami za pomocą odpowiednich magistral.
Urządzenia podłączone do magistrali mogą działać RÓWNOLEGLE. Swoje działania muszą synchronizować wówczas, gdy w tym samym czasie potrzebują dostępu do MAGISTRALI systemowej, czyli w trakcie przesyłania informacji do i z urządzeń. Taka równoległość zwiększa wydajność systemu.

MECHANIZM PRZERWAŃ

MECHANIZM PRZERWAŃ to mechanizm służący informowaniu o zdarzeniach zachodzących w urządzeniach wejścia-wyjścia. Takie zdarzenia to na przykład zakończenie zleconej operacji wejścia-wyjścia oraz zdarzenia zewnętrzne takie jak: naciśnięcie klawisza na klawiaturze, ruch myszki czy otrzymanie przez sieć pakietu danych. W momencie zajścia takiego zdarzenia urządzenie generuje tzw. sygnał przerwania wysyłany do procesora. W odpowiedzi procesor przerywa wykonywanie aktualnego ciągu instrukcji, zapisuje adres przerwania, wykonuje procedurę obsługi przerwania a następnie powraca do wykonywanego wcześniej ciągu instrukcji. Przypomina to wykonanie procedury w programie komputerowym z tą różnicą, że powodem wykonania jest urządzenie wejścia-wyjścia a nie sam program.

RODZAJE PRZERWAŃ

Przerwania dzielą się na dwa podstawowe rodzaje:

1. PRZERWANIA SPRZĘTOWE - przerwania wywoływane przez urządzenia wejścia-wyjścia
2. PRZERWANIA PROGRAMOWE - wykonywanie specjalnych instrukcji wywołujących procedurę obsługi przerwania

MECHANIZM ROZPOZNAWANIA PRZERWAŃ

ODPYTYWANIE polega na tym, że procedura obsługi przerwania odpytuje kolejno wszystkie urządzenia, czy to nie one spowodowały przerwanie; rozwiązanie to jest stosowane wówczas, gdy mamy tylko jeden rodzaj przerwań.
WEKTOR PRZERWAŃ - w systemie może być wiele rodzajów przerwań, a każdemu rodzajowi odpowiada osobna procedura obsługi; adresy tych procedur są przechowywane w tablicy systemowej nazywanej wektorem przerwań; rozwiązanie to jest bardziej efektywne niż odpytywanie.

CHARAKTER OBSŁUGI PRZERWAŃ

Procedura obsługi przerwania powinna mieć charakter atomowy, tzn. jej wykonanie nie powinno być przerywane, np. przez inne procedury obsługi przerwań. Nie można jednak zagwarantować, że w trakcie wykonywania procedury obsługi przerwania nie pojawi się żadne przerwanie sprzętowe. Dlatego też istnieje możliwość blokowania przerwań. Jeżeli przerwania są zablokowane, to obsługa pojawiających się przerwań sprzętowych jest odłożona do momentu odblokowania przerwań. Istnieje jednak wyjątek od tej zasady. Niektóre przerwania, tzw. przerwania nieblokowalne (niemaskowalne), sygnalizujące błędy krytyczne dla działania systemu komputerowego nie podlegają blokowaniu.

SYNCHRONICZNE I ASYNCHRONICZNE WEJŚCIE-WYJŚCIE

System operacyjny zlecając operację wejścia-wyjścia musi wiedzieć kiedy taka operacja zostanie wykonana. Stosowane są dwa rozwiązania:

- synchroniczne wykonywanie operacji wejścia-wyjścia - procesor zleca wykonanie operacji i czeka, aż otrzyma potwierdzenie zakończenia operacji
- asynchroniczne wykonywanie operacji wejścia-wyjścia - procesor zleca wykonanie operacji, przerywa wykonywanie aktualnie wykonywanego zadania i nie czekając na zakończenie się operacji wejścia-wyjścia zaczyna wykonywać inne zadanie; po zakończeniu operacji wejścia-wyjścia urządzenie zewnętrzne generuje przerwanie, a procedura obsługi przerwania odnotowuje ten fakt

SYNCHRONICZNE I ASYNCHRONICZNE WEJŚCIE-WYJŚCIE - WADY I ZALETY

Synchroniczne operacje wejścia-wyjścia są łatwiejsze do zaimplementowania, jednak operacje asynchroniczne pozwalają na zwiększenie wydajności systemu. W środowisku wieloprogramowym pozwalają one na równoczesne wykonywanie wielu operacji wejścia-wyjścia i obliczeń. Podobnie, jeżeli zadanie chce wykonać operację wejścia-wyjścia na urządzeniu, które aktualnie jest zajęte wykonywaniem innej operacji, to wykonanie takiego zadania jest wstrzymywane, aż do momentu, gdy dane urządzenie będzie dostępne i zlecona operacja zostanie wykonana. System operacyjny, który wykonuje operacje wejścia-wyjścia asynchronicznie musi pamiętać informacje o wszystkich wykonywanych i oczekujących operacjach wejścia-wyjścia. Informacje te są przechowywane w **tablicy stanów urządzeń**.

SYNCHRONICZNE I ASYNCHRONICZNE WEJŚCIE-WYJŚCIE - WADY I ZALETY

Tablica ta zawiera po jednej pozycji dla każdego urządzenia w systemie. Na pozycję w takiej tablicy składa się opis aktualnego stanu urządzenia oraz kolejka zleceń operacji wejścia-wyjścia. Jeżeli urządzenie nie wykonuje żadnej operacji wejścia-wyjścia, to odpowiadająca mu kolejka jest pusta. W przeciwnym przypadku, pierwsze zlecenie w kolejce odpowiada aktualnie wykonywanej operacji, a kolejne zlecenia odpowiadają operacjom oczekującym na wykonanie. Zlecenia w kolejce zawierają m.in. informacje o zadaniu, od którego pochodzi zlecenie, tak że w momencie wykonania zlecenia system operacyjny wie, które zadanie należy wznowić.

UKŁAD DMA I JEGO ROLA

W trakcie wykonywania asynchronicznej operacji wejścia-wyjścia następujące czynności zajmują czas pracy procesora: obsługa tablicy stanów urządzeń, zlecenie wykonania operacji sterownikowi urządzenia, przesłanie zapisywanych/odczytywanych danych, obsługa przerwania kończącego wykonanie operacji. Jeżeli zapisywane/odczytywane są pojedyncze znaki, tak jak w przypadku odczytywania znaków z klawiatury, to takie rozwiązanie jest akceptowalne. Jeśli jednak przesyłane są duże ilości danych, jak to ma miejsce w przypadku zapisu/odczytu z dysku, to procesor nie musi sam kopiować danych z/do pamięci operacyjnej do/z sterownika urządzenia. Może go w tym wyręczyć **układ DMA** (bezpośredniego dostępu do pamięci, ang. direct memory access), a procesor może w tym czasie wykonywać obliczenia. Układ DMA do przesyłania informacji potrzebuje oczywiście dostępu do magistrali systemowej. Korzysta z niej jednak w tych chwilach, gdy procesor z niej nie korzysta.

RODZAJE PAMIĘCI

W systemie komputerowym znajduje się wiele rodzajów pamięci. Jednym z podstawowych rodzajów pamięci jest **pamięć operacyjna**. Programy i dane, aby mogły być bezpośrednio przetwarzane przez procesor muszą być umieszczone w pamięci operacyjnej. Gdyby nie dwie jej wady, byłaby to pamięć idealna: jest ona zbyt mała, aby pomieścić wszystkie informacje, jakie muszą być pamiętane w systemie komputerowym, jest to pamięć ulotna, tzn. w momencie wyłączenia zasilania jej zawartość znika. Procesor przetwarzając informacje zawarte w pamięci operacyjnej przechowuje niektóre dane w **rejestrach**. Rejestry możemy traktować jak bardzo małą pamięć, do której procesor ma bardzo szybki dostęp.

RODZAJE PAMIĘCI

Istnieje jeszcze jeden rodzaj pamięci dostępnej procesorowi - **pamięć podręczna procesora**. Pamięć podręczna jest szybsza od pamięci operacyjnej, ale jest też od niej mniejsza. Zawiera ona kopię tych obszarów pamięci operacyjnej, z których aktualnie (najczęściej) korzysta procesor. Ponieważ pamięć podręczna dubluje funkcjonalność pamięci operacyjnej, jest ona niewidoczna z programistycznego punktu widzenia. Zastosowanie pamięci podręcznej procesora istotnie polepsza wydajność systemu.

RODZAJE PAMIĘCI

Większość systemów komputerowych jest wyposażona w masową pamięć trwałą, czyli taką, która nie traci swojej zawartości po odłączeniu zasilania. W pamięci tej przechowywane jest wykorzystywane oprogramowanie, najrozmaitsze dane, jak również programy tworzące system operacyjny. Najczęściej pamięć ta jest zrealizowana w postaci dysków magnetycznych, choć możliwe są też inne rozwiązania. Zwykle istnieje też możliwość podłączenia do systemu komputerowego wymiennych nośników pamięci trwałej: dyskietek, taśm magnetycznych, dysków CD/DVD itp. Nośniki te są wolniejsze od dysków magnetycznych, jednak ich zaletą jest wymienność, możliwość przenoszenia danych między systemami komputerowymi oraz niska cena.

RODZAJE PAMIĘCI

Dyski magnetyczne mają następującą konstrukcję: zestaw sztywnych dysków pokrytych dwustronnie nośnikiem magnetycznym osadzony jest na wspólnej osi i obraca się ze stałą dużą prędkością. Podobnie jak ramię gramofonu przesuwa się nad płytą gramofonową, tak nad powierzchniami dysków przesuwają się ramiona z głowicami magnetycznymi. Ramiona te są osadzone na wspólnej osi i znajdują się zawsze w tej samej pozycji jedno nad drugim. Informacja zapisana po jednej stronie dysku jest podzielona na ścieżki i sektory. Ścieżka ma kształt okręgu i po umieszczeniu głowicy nad ścieżką może być odczytana bez konieczności przesuwania głowicy. Ścieżka jest dalej podzielona na sektory - podstawowe jednostki zapisu/odczytu. Sektory mają zwykle wielkość 512B. Zestaw ścieżek, które mogą być odczytane przez wszystkie głowice, przy zadanym ich położeniu tworzy tzw. cylinder. Sposób rozmieszczenia informacji na dysku nie jest bez znaczenia.

RODZAJE PAMIĘCI

Informacje, które są zapisane w obrębie jednego cylindra mogą być odczytane stosunkowo szybko. Natomiast jeżeli odczytanie informacji wymaga przemieszczania głowic, to wiąże się to z dużo większym opóźnieniem. Porównując różne rodzaje pamięci w systemie komputerowym można zauważyć, że układają się one w hierarchię: od pamięci szybkich, drogich i niedużych, do pamięci wolniejszych, tanich i masowych. Na szczycie tej hierarchii znajdują się rejestry procesora, dalej jest pamięć podręczna procesora, pamięć operacyjna i pamięci dyskowe. Oczywiście w przypadku konkretnych systemów komputerowych hierarchia ta może być bogatsza.

BUFOROWANIE I WIRTUALIZACJA

W obrębie hierarchii pamięci system operacyjny realizuje dwa przeciwstawne mechanizmy: buforowanie i pamięć wirtualna. Mechanizm buforowania polega na tym, że zamiast przechowywać aktualnie wykorzystywane informacje w wolniejszej pamięci, przechowuje się je w wydzielonym obszarze szybszej pamięci. Za cenę zużycia części szybszej pamięci zyskujemy na wydajności. Przykładami buforowania są: buforowanie operacji dyskowych i pamięć podręczna procesora. Mechanizm pamięci wirtualnej polega na tym, że część informacji, które powinny być przechowywane w szybszej pamięci, ale nie są w danej chwili używane, są przechowywane w wolniejszej pamięci. Dopiero, gdy są faktycznie potrzebne, są przenoszone do szybszej pamięci. Za cenę pewnego wydłużenia czasu dostępu do informacji zyskujemy złudzenie, że dysponujemy obszerniejszą pamięcią szybką. Mimo, że mechanizmy te wydają się wzajemnie przeciwstawne, to współdziałają one w efektywniejszym wykorzystaniu systemu. Informacje, które są w danej chwili często wykorzystywane migrują do pamięci szybszych, a te, które w danej chwili nie są potrzebne są spychane do pamięci wolniejszych.

SPRZĘTOWE MECHANIZMY OCHRONY

Systemy operacyjne stanowią środowisko wykonywania dla wielu zadań pochodzących często od wielu użytkowników. Zadania te współdzielą między siebie rozmaite zasoby systemowe. Dlatego też system operacyjny pełni nie tylko rolę pomocniczą, ale i nadzorczą. Pilnuje, czy każde z zadań korzysta jedynie z tych zasobów, które zostały mu przydzielone i nie szkodzi innym zadaniom (takie szkodliwe działania nie muszą wynikać ze złośliwości programistów, lecz ze zwykłych błędów programistycznych). Aby pełnić funkcje nadzorcze, system operacyjny wymaga pewnych sprzętowych mechanizmów ochronnych.

SPRZĘTOWE MECHANIZMY OCHRONY

Jednym ze sprzętowych mechanizmów ochronnych jest dualny tryb pracy procesora. Polega on na tym, że procesor może pracować w dwóch trybach: **systemowym** (uprzywilejowanym) i **użytkownika** (ograniczonym). System operacyjny pracuje w trybie systemowym, a zadania użytkowników w trybie użytkownika. Procesor będąc w trybie systemowym może wykonywać wszystkie instrukcje, natomiast w trybie użytkownika wykonanie pewnych instrukcji jest zabronione. Instrukcje takie nazywamy **instrukcjami uprzywilejowanymi**.

SPRZĘTOWE MECHANIZMY OCHRONY

Zastanówmy się jakich instrukcji nie można wykonywać w trybie użytkownika. Oczywiście zmiana trybu pracy procesora jest instrukcją uprzywilejowaną. Jedyną metodą powrotu z trybu użytkownika do trybu systemowego jest przerwanie (sprzętowe lub programowe) - procedury obsługi przerwań są wykonywane w trybie systemowym. W konsekwencji, cały mechanizm przerwań pozostaje w gestii systemu. Tak więc blokowanie i odblokowywanie przerwań są również instrukcjami uprzywilejowanymi.

SPRZĘTOWE MECHANIZMY OCHRONY

Jednym ze współdzielonych przez zadania zasobów jest pamięć. Każde zadanie powinno mieć dostęp tylko do przydzielonej mu przez system operacyjny pamięci, a nie powinno mieć dostępu do obszarów pamięci innych zadań, ani do pamięci systemu operacyjnego. Tak więc, w trybie użytkownika przy każdym dostępie do pamięci trzeba sprawdzić, czy aktualnie wykonywane zadanie ma prawo odwoływania się do danej komórki pamięci. Jest to realizowane przez **sprzętowy mechanizm adresowania pamięci**.

SPRZĘTOWE MECHANIZMY OCHRONY

Urządzenia wejścia-wyjścia są również współdzielone przez zadania użytkowników. Nadzór nad ich wykorzystaniem należy do systemu operacyjnego. Dlatego też instrukcje komunikacji z urządzeniami zewnętrznymi (ich sterownikami) są instrukcjami uprzywilejowanymi. Dotyczy to również układu DMA i innych programowalnych mechanizmów sprzętowych, nawet jeśli nie są one urządzeniami wejścia-wyjścia.

SPRZĘTOWE MECHANIZMY OCHRONY

Kolejnym zasobem współdzielonym przez zadania jest sam procesor. Jeżeli zadanie zleci wykonanie operacji wejścia-wyjścia (lub ogólniej, wywoła jakąkolwiek funkcję systemową), to system może wstrzymać jego wykonywanie i przydzielić procesor innemu zadaniu. Jeżeli jednak zadanie jest zajęte wykonywaniem obliczeń, to w jaki sposób można zapobiec zawłaszczeniu przez nie dostępu do procesora? Jest to zrealizowane za pomocą **mechanizmu przerwań zegarowych**. W systemie komputerowym działa programowalny zegar, który z określoną częstotliwością generuje przerwanie sprzętowe. Jeżeli zadanie powinno przekazać dostęp do procesora innemu zadaniu, to przy okazji najbliższego przerwania zegarowego procedura obsługi przerwania może wstrzymać wykonywanie zadania i uruchomić inne zadanie. Oczywiście programowanie zegara stanowi instrukcję uprzywilejowaną. Do instrukcji uprzywilejowanych należy też kilka instrukcji, o których nie było tu mowy, np. instrukcja zatrzymania systemu komputerowego halt.

ROLA PRZERWAŃ W SYSTEMIE OPERACYJNYM

Spróbujmy podsumować wszystkie sytuacje, w których w systemie operacyjnym występują przerwania. Sygnalizują one zakończenie operacji wejścia-wyjścia. Informują o zewnętrznych zdarzeniach i komunikacji docierającej do systemu komputerowego. Przerwania programowe, dzięki temu, że powodują przejście w tryb systemowy, są zwykle wykorzystywane jako forma wywoływania funkcji systemowych. Można więc śmiało powiedzieć, że wszystkim zdarzeniom w systemie operacyjnym towarzyszą przerwania. Inaczej mówiąc: system operacyjny jest sterowany przerwaniami (ang. event-driven).

SKŁADOWE SYSTEMU OPERACYJNEGO

System operacyjny to duży i bardzo złożony system informatyczny. Systemy takie muszą mieć modularną strukturę, inaczej trudno poradzić sobie z ich złożonością. W typowym systemie operacyjnym wyróżniamy następujące podsystemy:

1. Podsystem zarządzania procesami
2. Podsystem zarządzania pamięcią
3. Podsystem systemu plików
4. Podsystem wejścia-wyjścia
5. Podsystem pamięci pomocniczej
6. Podsystem usług sieciowych
7. Podsystem ochrony
8. Interpreter poleceń i programy użytkowe

PODSYSTEM ZARZĄDZANIA PROCESAMI

Na początku musimy pokrótce przedstawić pojęcie procesu. **Proces** to wykonujący się program. Program to obiekt pasywny (tak jak plik na dysku), a proces to obiekt aktywny (uruchomiony program). W szczególności jeżeli równocześnie kilka razy uruchomimy ten sam program, to będziemy mieli do czynienia z kilkoma procesami, ale z tylko z jednym programem. Do tej pory używaliśmy pojęcia zadania, jest to jednak pojęcie bardziej ogólne. Jeżeli wykonanie zadania polega na uruchomieniu programu i przetworzeniu konkretnych danych, to możemy używać tych pojęć zamiennie. Zadanie może jednak obejmować kilka współbieżnie działających, współpracujących ze sobą procesów. Wówczas są to dwa różne pojęcia. Możemy też mieć do czynienia z procesami systemowymi, które nie przynależą do żadnych zadań - np. proces buforujący drukowane dokumenty.

PODSYSTEM ZARZĄDZANIA PROCESAMI

Podsystem zarządzania procesami śledzi wszystkie procesy działające w systemie (zarówno systemowe, jak i procesy użytkowników), przydzielone im zasoby, ich stan oraz to, który proces w danej chwili jest wykonywany. Podsystem ten odpowiada za:

- tworzenie i usuwanie procesów
- szeregowanie, wstrzymywanie i wznawianie procesów
- mechanizmy synchronizacji i komunikacji między procesami, oraz obsługi zakleszczeń

PODSYSTEM ZARZĄDZANIA PAMIĘCIĄ

Podsystem ten śledzi, które obszary pamięci operacyjnej są wolne, a które są zajęte i przez kogo. Przydziela on pamięć procesom i zwalnia ją. Zajmuje się on również pamięcią wirtualną. W środowisku wieloprogramowym decyduje on też o tym, które programy mają być załadowane do pamięci operacyjnej, a które mają oczekiwać na uruchomienie.

PODSYSTEM SYSTEMU PLIKÓW

Podsystem ten realizuje pojęcia plików i katalogów, ukrywając jednocześnie przed użytkownikiem sposób ich realizacji. Realizuje on:

- podstawowe operacje na plikach i katalogach
- dostęp do plików
- rozmieszczenie treści plików w pamięci pomocniczej
- składowanie plików na urządzeniach pamięci trwałej

PODSYSTEM WEJŚCIA-WYJŚCIA

Podsystem ten ukrywa przed użytkownikiem szczegóły dotyczące urządzeń wejścia/wyjścia. W jego skład wchodzą:

- moduły sterujące poszczególnych urządzeń, ukrywające szczegóły charakterystyczne dla poszczególnych urządzeń
- ujednolicony interfejs modułów sterujących urządzeniami
- moduł zarządzający buforowaniem i czytaniem z wyprzedzeniem pamięci pomocniczej

PODSYSTEM PAMIĘCI POMOCNICZEJ

Podsystem ten zarządza urządzeniami pamięci pomocniczej (zwykle jest to pamięć dyskowa), śledzi, które obszary są wolne, a które zajęte, przydziela pamięć, szereguje odwołania do pamięci pomocniczej. Podsystem ten świadczy usługi zarówno podsystemowi systemu plików, jak i pamięci wirtualnej.

PODSYSTEM USŁUG SIECIOWYCH

System rozproszony to system składający się z wielu komputerów połączonych siecią komputerową. Pewne funkcje takiego systemu nie są realizowane w obrębie pojedynczego komputera, lecz są rozproszone między połączone komputery. Podsystem ten realizuje komunikację między połączonymi komputerami oraz rozproszone funkcje systemu. Przykładem takich rozproszonych funkcji może być sieciowy system plików.

POSYSTEM OCHRONY

W systemie, z którego korzysta wielu użytkowników, istnieje potrzeba określenia z jakich zasobów mogą korzystać poszczególni użytkownicy i w jakim zakresie. Zajmuje się tym właśnie podsystem ochrony.

W skład systemu operacyjnego wchodzi również interpreter poleceń systemowych, który stanowi dla użytkowników interfejs do wydawania poleceń systemowi operacyjnego. W poprzednich wykładach przedstawiliśmy przykładowy interpreter poleceń bash. Interpreter poleceń staje się potężnym narzędziem dopiero, gdy korzysta się z niego w połączeniu z rozmaitymi programami użytkowymi dostarczanymi w ramach systemu operacyjnego.

USŁUGI SYSTEMU OPERACYJNEGO

System operacyjny tworzy środowisko udostępniające pewne usługi użytkownikom oraz ich programom. Można powiedzieć, że system operacyjny udostępnia dwa interfejsy. Jeden z nich, przeznaczony dla programów użytkowników, tworzą funkcje systemowe. Drugi z nich, przeznaczony dla użytkowników, tworzą programy systemowe oraz systemowy interfejs użytkownika. Funkcjonalność tych dwóch interfejsów zachodzi na siebie, choć nie jest identyczna. Usługi systemowe udostępniane w ramach tych dwóch interfejsów można pogrupować w poniższy sposób.

Uruchamianie programów.
Jedno z podstawowych poleceń, jakie można zlecić systemowi operacyjnemu, to wykonanie wskazanego programu.
Operacje wejścia-wyjścia.
Wykonywane programy pobierają dane i przekazują wyniki za pomocą operacji wejścia/wyjścia. System operacyjny musi więc realizować i udostępniać takie operacje.
Operacje na systemie plików.
Operacje wejścia-wyjścia powinny być uzupełnione operacjami pozwalającymi manipulować plikami i katalogami.

Przydzielanie zasobów.
Grupa ta obejmuje przydzielanie i zwalnianie najrozmaitszych zasobów systemowych.

Komunikacja.
Grupa ta obejmuje zarówno komunikację między procesami w obrębie jednej maszyny, komunikację między procesami poprzez sieć komputerową oraz usługi komunikacyjne dla użytkowników.

Ochrona.
System operacyjny powinien kontrolować wykorzystanie różnych zasobów przez różne procesy/użytkowników. Aby dostosować ten mechanizm kontroli do potrzeb użytkowników, system operacyjny musi również udostępniać operacje służące do ustalania, kto ma prawo z czego korzystać i w jakim zakresie.

Wykrywanie błędów.
System operacyjny, to taki rodzaj systemu informatycznego, w którym w bardzo wielu sytuacjach mogą zaistnieć błędy. Wszystkie takie błędy powinny być wykrywane i sygnalizowane w sposób ułatwiający zlokalizowanie ich przyczyny.

Rozliczenia.
System operacyjny prowadzi pewne statystyki wykorzystania zasobów systemowych przez poszczególnych użytkowników (np. czasu pracy procesora, czy wykorzystanej przestrzeni dyskowej).

FUNKCJE SYSTEMOWE

Programy użytkowników mają dostęp do usług systemu operacyjnego poprzez tzw. funkcje systemowe. Funkcje te są widoczne w języku programowania podobnie jak funkcje z tego języka, a ich konkretny mechanizm wywoływania jest ukryty przed programistą. Czasami też w języku dostępne są operacje trochę wyższego poziomu, które są zaimplementowane za pomocą funkcji systemowych. Na przykład, dostępne operacje na plikach mogą być niskopoziomowe - pliki traktowane są w nich tak jak widzi je system operacyjny - lub mogą implementować pewną strukturę pliku - np. to, że plik jest ciągiem rekordów - która nie jest znana systemowi operacyjnemu. Funkcje udostępniane przez system operacyjny możemy pogrupować w następujący sposób.

PODZIAŁ FUNKCJI SYSTEMOWYCH

Operacje na procesach.
Ta grupa funkcji obejmuje m.in.: tworzenie nowych procesów, uruchamianie programów, zakończenie się procesu, przerwanie działania innego procesu, pobranie informacji o procesie, zawieszenie i wznowienie procesu, oczekiwanie na określone zdarzenie, przydzielanie i zwalnianie pamięci.

Operacje na plikach.
Ta grupa funkcji obejmuje takie operacje, jak: utworzenie pliku, usunięcie pliku, otwarcie pliku, odczyt z pliku, zapis do pliku, pobranie lub zmiana atrybutów pliku.

Operacje na urządzeniach.
Zamówienie i zwolnienie urządzenia, odczyt z i zapis do urządzenia, pobranie lub zmiana atrybutów urządzenia, (logiczne) przyłączenie lub odłączenie urządzenia.

Informacje systemowe.
Funkcje te obejmują pobieranie i/lub ustawianie najrozmaitszych informacji systemowych, w tym: czasu i daty, wersji systemu operacyjnego, atrybutów użytkowników itp.
Komunikacja.
Ta grupa funkcji obejmuje przekazywanie informacji między procesami (w obrębie jednego komputera lub poprzez sieć komputerową). Spotykane są dwa schematy komunikacji: przekazywanie komunikatów i pamięć współdzielona.
Przekazywanie komunikatów polega na tym, że jeden proces może wysłać (za pośrednictwem systemu operacyjnego) pakiet informacji, który może być odebrany przez drugi proces. Mechanizm pamięci współdzielonej polega na tym, że dwa procesy uzyskują dostęp do wspólnego obszaru pamięci. Cokolwiek jeden z procesów zapisze w tym obszarze, będzie widoczne dla drugiego procesu.

PROGRAMY SYSTEMOWE

Programy systemowe są to programy rozprowadzane razem z systemem, które umożliwiają użytkownikowi dostęp do pewnych usług systemowych. Niektóre z nich są tylko interfejsem do funkcji systemowych, niektóre łączą proste funkcje systemowe w bardziej złożone usługi, a niektóre implementują usługi niedostępne w postaci funkcji systemowych. Programy systemowe można podzielić na następujące kategorie.

RODZAJE PROGRAMÓW SYSTEMOWYCH

Operacje na plikach.
Programy te umożliwiają kopiowanie, usuwanie, przemianowywanie, przemieszczanie, archiwizowanie oraz przeglądanie plików i katalogów.
Tworzenie i modyfikacja plików.
Programy te umożliwiają tworzenie, edycję i przetwarzanie plików (np. sortowanie czy filtrowanie informacji zawartych w pliku).
Kompilatory i interpretery.
System operacyjny zwykle dostarcza środowiska programistycznego przynajmniej dla jednego języka programowania (tego, w którym napisany jest sam system). Często udostępnia też interpretery umożliwiające łatwe tworzenie i uruchamianie zadań (wliczając w to interpreter poleceń systemowych). Wszystkie programy systemowe tworzące takie środowiska programistyczne zaliczamy do tej kategorii.

Uruchamianie programów.
Uruchamianie programów, choć nie jest to zwykle widoczne dla użytkownika, może wymagać obecności pewnych programów scalających, ładujących, czy interpretujących kod określonej postaci. Programy te zaliczamy do niniejszej kategorii. Czasami to, czy dany program przynależy do tej, czy do poprzedniej kategorii zależy od postaci kodu wynikowego. Ponieważ system operacyjny może wykonywać różne rodzaje kodów, może się zdarzyć, że ta kategoria będzie zachodzić na poprzednią.

Komunikacja.
Do tej kategorii zaliczają się programy umożliwiające komunikację między różnymi użytkownikami i/lub komputerami, np. poczta elektroniczna, zdalna sesja robocza czy kopiowanie plików między komputerami.

Informacje systemowe.
Programy te prezentują w czytelnej postaci informacje systemowe i ewentualnie umożliwiają ich modyfikację.

STRUKTURA SYSTEMU OPERACYJNEGO

System operacyjny, to bardzo złożony system informatyczny. Musi on mieć odpowiednią strukturę, tak aby jego zadania mogły być podzielone na mniejsze moduły o dobrze określonej funkcjonalności i interfejsach. Niestety nie zawsze tak jest. Czasami systemy operacyjne powstawały jako niewielkie systemy o ograniczonej funkcjonalności, pozbawione właściwej struktury, a z czasem rozrastały się do systemów operacyjnych, ciągnąc ze sobą bagaż złych decyzji projektowych.

Przykładem takiego systemu może być MS-DOS. Można w nim wyróżnić cztery moduły: **BIOS** (ang. basic input-output system) - moduły obsługi urządzeń zapisane w pamięci stałej ROM (ściśle rzecz biorąc nie wchodzą one w skład MS-DOS, lecz są przezeń wykorzystywane), **Moduł obsługi urządzeń**, **Rezydentna część systemu i programy rezydentne**, **Interpreter poleceń i programy użytkownika**. Jeden z problemów polega na tym, że moduły te nie tworzą kolejnych "warstw", poziomów abstrakcji. Programy rezydentne i programy użytkownika mogą równie dobrze bezpośrednio komunikować się ze sterownikami urządzeń, korzystać z modułów BIOS, jak i korzystać z funkcji systemowych. W takiej sytuacji nie można zapewnić należytej ochrony w systemie i nie można tego poprawić nie zmieniając struktury systemu.

System **Unix** ma inną strukturę. Składa się on z jądra i programów systemowych. Programy systemowe znajdują się powyżej interfejsu funkcji systemowych, a jądro poniżej tego interfejsu. Można więc powiedzieć, że Unix składa się z warstw. Jednakże funkcjonalność zebrana w jądrze może wydawać się nadmierna.

Trochę inne podejście zastosowano w systemie **MAC OS**. Jądro systemu zostało tam zredukowane do minimum, a wszystkie pozostałe funkcje zostały przeniesione do programów systemowych, lub wręcz programów użytkowych. W tym przypadku mówimy o architekturze mikrojądra. Pewnym rozwiązaniem problemu modularyzacji systemu operacyjnego jest tzw. podejście warstwowe. W podejściu tym dzielimy system operacyjny na warstwy. Każda warstwa udostępnia warstwie powyżej pewien interfejs. Implementacja danej warstwy może korzystać z operacji udostępnianych przez warstwę poniżej.

Zwykle część operacji udostępnianych przez daną warstwę polega na wywoływaniu operacji niższej warstwy, a część jest zaimplementowanych w danej warstwie. Sprzęt możemy traktować jako warstwę najniższą, a środowisko pracy użytkownika, jako interfejs warstwy najwyższej. Pewną wadą podejścia warstwowego jest zmniejszona efektywność wywoływania operacji implementowanych przez niższe warstwy, wynikająca z przekazywania wywołania operacji między kolejnymi warstwami.