

Procesor – budowa, schemat, tryby pracy

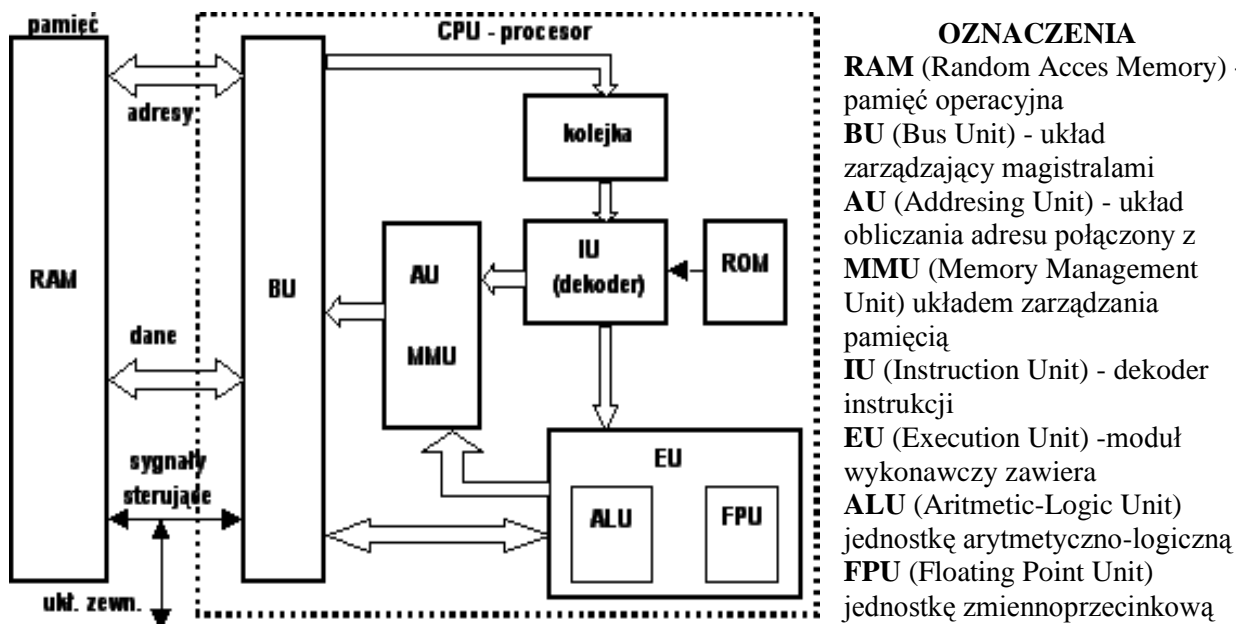
Procesor (ang. *processor*), także CPU (ang. *Central Processing Unit*) – to główny element komputera, **urządzenie cyfrowe sekwencyjne**, którego zadaniem jest wykonywanie rozkazów i sterowanie pracą wszystkich pozostałych bloków systemu (m.in. pamięci i układów wejścia-wyjścia). Procesor przetwarza informacje, wykonując na niej elementarne operacje zwane instrukcjami maszynowymi (bądź rozkazami). Ciąg takich instrukcji realizujący konkretne zadanie przetwarzania informacji nazywamy programem. Do systemu mikroprocesorowego oprócz danych wejściowych musimy więc dostarczyć także program lub zestaw programów, czyli oprogramowanie (ang. *software*). W przypadku systemu mikroprocesorowego sposób przetwarzania informacji jest określony głównie przez oprogramowanie. Ułatwia to w razie potrzeby zmianę sposobu przetwarzania informacji.

Procesory (zwane mikroprocesorami) wykonywane są zwykle jako układy scalone zamknięte w hermetycznej obudowie, często posiadającej złocone wyprowadzenia (stosowane ze względu na odporność na utlenianie). Ich sercem jest **monokryształ krzemu**, na który **naniesiono techniką fotolitografii szereg warstw półprzewodnikowych**, tworzących, w zależności od zastosowania, sieć od kilku tysięcy do kilkuset milionów tranzystorów. Połączenia wykonane są z metalu (aluminium, miedź).

Wszystkie mikroprocesory zawierają podobne elementy:

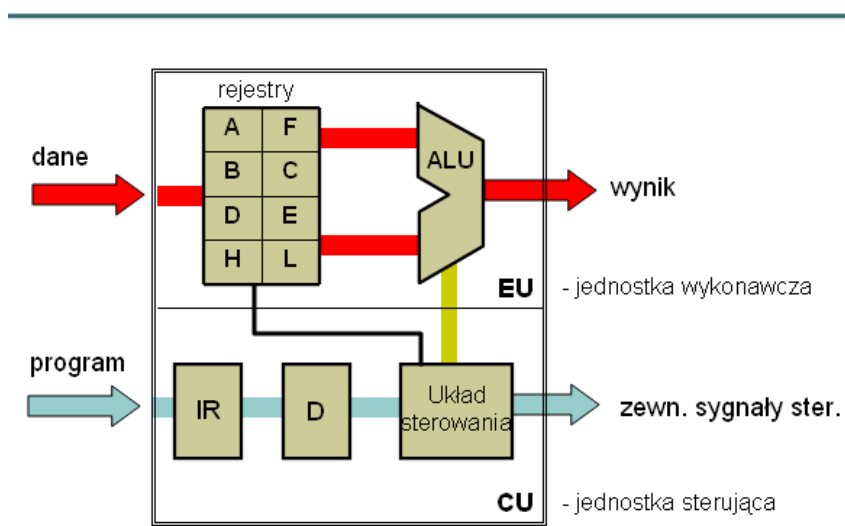
1. **układ sterowania i synchronizacji**, który kontroluje pracę procesora i wytwarza sygnały potrzebne do sterowania niektórymi elementami komputera.
2. **arytmometr**, czyli układ, który wykonuje operacje arytmetyczne i logiczne (niektóre procesory mają kilka arytmometrów).
3. **rejstry**, tj. układy pamięci.
4. wbudowana **pamięć podręczna cache**, która działa podobnie do zewnętrznej pamięci RAM. **Zapewnia ona, że procesor nie jest zmuszony czekać na dane potrzebne mu do pracy.**
5. **koprocesor matematyczny**, który jest zestawem instrukcji przeznaczonych do **obsługi skomplikowanych operacji matematycznych**.
6. wewnętrzne szyny łączące elementy procesora.

Schemat blokowy procesora 2:



ALU - układ ten jest często nazywany układem wykonawczym procesora, ponieważ wykonuje on większość rozkazów zleczanych procesorowi. Blok ALU jest układem, który realizuje różne funkcje w zależności od zaprogramowanej operacji, tj. rozkazu umieszczonego w programie. Argumentami tych rozkazów są słowa binarne. Rozkazy te mogą dotyczyć operacji arytmetycznych (dodawanie i odejmowanie) lub logicznych (sumowanie, mnożenie itp.).

Schemat blokowy mikroprocesora



Rejestry:

Zarówno jednostka arytmetyczno-logiczna jak i układ sterowania musi współpracować z określonym zestawem rejestru. Zawartość pewnej części rejestru może być zmieniona w wyniku wykonania przez procesor określonej instrukcji. Rejestry takie nazywamy rejestrami dostępnymi programowo. W grupie tych rejestrów występują takie typy rejestrów, których

odpowiedniki znajdują się praktycznie w każdym mikroprocesorze. Ich pojemność lub ilość może się zmieniać jednak zastosowanie i wykonywane zadania pozostają takie same.



Oznaczenia:

- A - akumulatory
- B, C, D, E, H, L - rejestry robocze
- SP - wskaźnik stosu
- F - rejestr flagowy
- PC - licznik rozkazów

- **Akumulator - A** - jest to rejestr, który zawiera jeden z argumentów wykonywanej operacji i do którego ładowany jest wynik wykonywanej operacji.
- **Rejestr flagowy- F** - zawiera dodatkowe cechy wyniku wykonywanej operacji, które potrzebne są do podjęcia decyzji o dalszym sposobie przetwarzania informacji. Cechami tymi mogą być: znak wyniku, przekroczenie zakresu, parzysta lub nieparzysta liczba jedynek. Wystąpienie określonej cechy sygnalizowane jest ustawieniem lub wyzerowaniem określonego bitu w rejestrze flagowym. Ustawione bity nazywane są znacznikami lub flagami.
- **Licznik rozkazów - PC**- jest jednym z istotniejszych rejestrów dzięki któremu procesor potrafi pobierać kolejne rozkazy do wykonania. Licznik rozkazów zawiera adresy komórki pamięci w której przechowywany jest kod rozkazu przeznaczony do wykonania jako następny.
- Oprócz tego procesor ma kilka (kilkanaście) rejestrów używanych w czasie wykonywania niektórych rozkazów np. **wskaźnik stosu – SP** – służący do adresowania pamięci. **Stosem nazywamy** wyróżniony obszar pamięci, używany według następujących reguł:
 - Informacje zapisywane są na stos do kolejnych komórek przy czym żadnego adresu nie wolno pominąć
 - Informacje odczytuje się w kolejności odwrotnej do zapisu
 - Informacje odczytujemy z ostatnio uzupełnianej komórki natomiast zapisujemy do pierwszej wolnej
 - Stos jest wydzielonym miejscem w pamięci w którym obowiązuje zasada: ostatni wchodzi pierwszy wychodzi.

Procesor posiada określone parametry:

- rozmiar elementów budujących jego strukturę – im są one mniejsze, tym niższe jest zużycie energii, napięcie pracy oraz wyższa możliwa do osiągnięcia częstotliwość pracy,
- długość słowa (liczba bitów) danych wewnątrz procesora (w mikroprocesorach 8, 16 32 lub 64 bity),

- długość słowa adresu (zwykle 16 lub 20 bitów),
- lista rozkazów (kopiowanie danych, działania arytmetyczne, działania na bitach, skoki),
- czas cyklu maszynowego – szybkość z jaką procesor wykonuje rozkazy; przy danej architekturze procesora, szybkość ta w znacznym stopniu zależy od czasu trwania pojedynczego taktu.

Przykładowymi producentami procesorów są firmy: Intel, Celeron, AMD, Cyrix, Motorola, Via.

Tryby pracy procesora:

Tryb pracy procesora decyduje, w jaki sposób zarządza on pamięcią systemową i zadaniami do wykonania. Można wyróżnić następujące trzy tryby pracy:

- **tryb rzeczywisty (oprogramowanie 16-bitowe),**
- **tryb chroniony (oprogramowanie 32-bitowe),**
- **tryb wirtualny (oprogramowanie 16-bitowe uruchamiane na platformie 32-bitowej).**

Tryb rzeczywisty:

Oryginalny komputer IBM PC był wyposażony w procesor 8088, który przy użyciu 16-bitowych rejestrów wewnętrznych potrafił wykonywać instrukcje 16-bitowe oraz za pośrednictwem 20 linii adresowych współpracować z pamięcią o pojemności maksymalnej 1MB. Oprogramowanie, w które był wyposażony oryginalny IBM PC zostało stworzone z myślą o współpracy z procesorem 8088, dlatego też wykorzystywało zestaw instrukcji 16-bitowych i 1-megabajtowy model pamięci. Przykładowo, system DOS i wszystkie programy dla niego przeznaczone oraz system Windows w wersjach 1.x - 3.x wraz z aplikacjami zostały napisane przy wykorzystaniu instrukcji 16-bitowych. 16-bitowe systemy operacyjne i aplikacje zostały stworzone dla oryginalnego procesora 8088.

Następne procesory, takie jak 286, nadal mogły wykonywać te same 16-bitowe instrukcje co procesor 8088, ale już znacznie szybciej. Inaczej mówiąc, procesor 286 był w pełni z nim kompatybilny, dlatego też można było na nim uruchamiać wszystkie programy 16-bitowe co w przypadku procesora 8088, ale oczywiście działały one o wiele szybciej. **Tryb wykonywania przez procesory 8088 i 286 instrukcji 16-bitowych został określony terminem trybu rzeczywistego (ang. real mode).** Wszystkie programy pracujące w trybie rzeczywistym mogą korzystać tylko z instrukcji 16-bitowych i współpracować z 20-bitową architekturą pamięci (1 MB). Tego typu oprogramowanie z reguły jest jednozadaniowe, co oznacza, że **jednocześnie może być uruchomiony tylko jeden program.** Ponadto nie ma wbudowanej żadnej ochrony zapobiegającej wykorzystaniu przez program obszaru pamięci już przydzielonego innemu programowi lub systemowi operacyjnemu. **Jeśli zatem uruchomiono więcej niż jeden program, dowolny z nich może spowodować zawieszenie całego systemu.**

Tryb chroniony:

Pierwszym procesorem 32-bitowym przeznaczonym dla komputerów PC i wprowadzonym do sprzedaży był **układ 386**. Procesor ten mógł wykonywać zupełnie nowy zestaw instrukcji 32-bitowych. **Aby w pełni wykorzystać taką możliwość, konieczne było stworzenie odpowiedniego 32-bitowego systemu operacyjnego i aplikacji.** Nowy, 32-bitowy tryb pracy procesora został określony terminem trybu chronionego (ang. *protected mode*). Nazwa wywodzi się stąd, że **programy działające w tym trybie otrzymują własny obszar pamięci, przez co nie dochodzi między nimi do konfliktów.** Tego typu ochrona sprawia, że system jest o wiele bardziej stabilny, ponieważ programy zawierające błąd nie mogą w prosty sposób zaszkodzić innym programom lub systemowi operacyjnemu. **Poza tym program, który się zawiesił, może być usunięty z pamięci bez szkody dla reszty systemu.**

Jest to zatem tryb pracy mikroprocesorów serii x86 wprowadzony w mikroprocesorze Intel 80286. **Tryb chroniony umożliwia adresowanie pamięci w większym zakresie niż 1 MB** (tryb rzeczywisty), wprowadza wiele **nowych udogodnień wspierających wielozadaniowość**, takich jak: sprzętowa ochrona pamięci (układ MMU), wsparcie przełączania kontekstu procesora i wiele innych.

Większość nowoczesnych systemów operacyjnych wykorzystuje procesory serii x86 właśnie w trybie chronionym. Zaliczają się do nich m.in.: Linux, Windows w wersji 3.0 i wyższych, systemy z rodziny BSD.

**Tryb chroniony został wprowadzony do mikroprocesora Intel 80286 wraz z trybem rzeczywistym (ang. real mode) dla zachowania zgodności z poprzednimi mikroprocesorami rodziny x86. W mikroprocesorze Intel 80286 w trybie chronionym 16-bitowe rejestry segmentowe CS, SS, DS, ES zawierają selektory, które jednoznacznie określają deskryptor (8-bajtowa struktura w pamięci) opisujący segment. Deskryptor dysponuje następującymi informacjami o segmencie: fizyczna lokalizacja segmentu w pamięci (24-bitowy adres bazowy, 16-bitowa długość segmentu w bajtach), prawa dostępu do segmentu i numer uprawnienia segmentu. W odróżnieniu od trybu chronionego w trybie rzeczywistym rejestry segmentowe CS, SS, DS, ES zawierają adresy bazowe (początków) segmentów. Procesor zaczyna pracę w trybie chronionym po ustawieniu bitu PE (ang. Protection Enable) na jeden, wyzerowanie tego bitu powoduje powrót do trybu rzeczywistego (bit PE to najmłodszy bit rejestru CR0 (ang. Control Register)).*

Tryb chroniony jest właściwym trybem pracy procesorów rodziny IA-32. Zapewnia on największą wydajność systemu; są w nim dostępne wszystkie instrukcje i mechanizmy procesora. Praca w tym trybie jest zalecana dla wszystkich nowych aplikacji i systemów operacyjnych. Dla zapewnienia zgodności programowej, bez utraty kontroli nad całym systemem (ochrona zasobów, wielozadaniowość itp.), wprowadzono możliwość uruchamiania programów trybu rzeczywistego bez potrzeby przełączania trybu pracy na rzeczywisty. Ta właściwość jest określana jako tryb wirtualny, i często jest nazywana kolejnym trybem pracy procesora. Tryb wirtualny jest właściwie 'pod-trybem' trybu chronionego i może być włączony poprzez ustawienie odpowiedniej flagi przez dowolne zadanie.

Tryb wirtualny:

Tryb wirtualny (zwany także **V86 lub Virtual 8086**) — to specjalny tryb pracy procesorów o architekturze IA-32, dostępny w trybie chronionym, który umożliwia

uruchamianie programów przeznaczonych dla trybu rzeczywistego. W trybie wirtualnym symulowane jest działanie analogiczne dla procesora Intel 8086 (faktycznie można uruchamiać kod także dla 8088, 80186 i 80188), tzn. otrzymuje dostęp do 1 MB pamięci i rejestrów procesora i może wykonywać te rozkazy, które mają sens w takim otoczeniu.

Tryb ten bywa nazywany wirtualnym trybem rzeczywistym. Jednak w odróżnieniu od "prawdziwego" trybu rzeczywistego, wszelki dostęp do portów procesora jest sankcjonowany przez system operacyjny pracujący w trybie chronionym.

W trybie wirtualnym procesor wykorzystuje swoje pełne możliwości. Używa 24 bitów adresu, co pozwala zaadresować 16 MB fizycznej pamięci. Ponadto dostępne są sprzętowe mechanizmy wspomagające obsługę pamięci wirtualnej, pracy wielozadaniowej i ochrony zasobów.

Tryb ten jest kluczem do zgodności 32-bitowego środowiska Windows i właściwie jest trybem wirtualnym środowiska 16-bitowego działającego w 32-bitowym trybie chronionym. Po uruchomieniu w systemie Windows sesji DOS jest wykorzystywany tryb wirtualny. Dzięki wielozadaniowości oferowanej przez tryb chroniony właściwie możliwe jest uruchomienie kilku sesji w trybie rzeczywistym, **w których działają programy na zasadzie pracy w wirtualnych komputerach.** Programy te mogą działać jednocześnie, nawet gdy są uruchomione aplikacje 32-bitowe.

**Należy zauważyć, że dowolny program działający w trybie wirtualnym ma dostęp tylko do 1 MB pamięci, co spowoduje, że będzie "przekonany o tym", że jest to pierwszy i jedyny megabajt pamięci dostępnej w systemie. Inaczej mówiąc, jeśli aplikacja napisana dla systemu DOS zostanie uruchomiona w trybie wirtualnym, będzie miała do dyspozycji tylko 640 kB. Takie ograniczenie wynika stąd, że w przypadku 16-bitowego środowiska pracy jest dostępny tylko 1 MB pamięci RAM, z czego 384 kB jest zarezerwowane dla samego systemu. Tryb wirtualny w pełni emuluje środowisko pracy procesora 8088, dlatego też aplikacje działają tak, jakby były uruchomione w komputerze oferującym tylko faktyczny tryb rzeczywisty. Każda wirtualna maszyna korzysta z własnej 1-megabajtowej przestrzeni adresowej, w której znajduje się obraz rzeczywistych funkcji systemu BIOS odwołujących się do urządzeń i emuluje wszystkie pozostałe rejestry oraz funkcje występujące w trybie rzeczywistym.*

Tryb wirtualny jest wykorzystywany w momencie uruchomienia, w oknie sesji DOS, 16-bitowego programu stworzonego dla systemu DOS lub Windows 3.x. Po uruchomieniu takiego programu system Windows tworzy wirtualną maszynę DOS, która przejmuje jego obsługę.

Wielozadaniowość – Multitasking, która cechuje systemy Windows, pozwala na współbieżne wykonywanie wielu procesów na raz.

Zwykle za poprawną realizację wielozadaniowości odpowiedzialne jest jądro systemu operacyjnego.

Wielozadaniowość zapewniona jest między innymi przez planistę, czyli część systemu operacyjnego realizującą algorytm szeregowania zadań w kolejce do przyznania czasu procesora.

Równoczesność jest pozorna. Dla uzyskania wrażenia wykonywania wielu zadań jednocześnie, konieczne staje się dzielenie czasu.

Systemy wielozadaniowe można podzielić na oferujące i nie oferujące wywłaszczania

W systemach z wywłaszczaniem może nastąpić przerwanie wykonywania procesu, odebranie mu procesora i przekazanie sterowania do planisty.

Pełne wywłaszczanie zapewniają mechanizmy sprzętowe działające niezależnie od oprogramowania (na przykład przerwanie zegarowe).

W systemach bez wywłaszczania procesy powinny same dbać o sprawiedliwy podział czasu, co często uzyskuje się pośrednio - proces dokonując wywołania systemowego, oddaje sterowanie procesowi jądra, lub jednemu z procesów systemowych i w ten sposób zrzeka się procesora. Program nie wykonywany pozostaje "w uśpieniu" do momentu, gdy znów zostanie mu przydzielony czas procesora.

Systemami wielozadaniowymi są:

- systemy z rodziny Uniksów (np. Linux)
- systemy z rodziny Microsoft Windows (np. Microsoft Windows 95)
- Mac OS i jego następcę Mac OS X

Nie jest:

- DOS (może być więcej niż jeden proces, ale tylko jeden będzie wykonywany)

Proces - to jedno z najbardziej podstawowych pojęć w informatyce. Z definicji jest to po prostu egzemplarz wykonywanego programu. Należy odróżnić jednak proces od wątku - każdy proces posiada własną przestrzeń adresową, natomiast wątki posiadają wspólną sekcję danych.

Wątek (ang. thread) - to jednostka wykonawcza w obrębie jednego procesu, będąca kolejnym ciągiem instrukcji wykonywanym w obrębie tych samych danych (w tej samej przestrzeni adresowej).

Wątki tego samego procesu korzystają ze wspólnego kodu i danych, mają jednak oddzielne stosy.

W systemach wieloprocessorowych, a także w systemach z wywłaszczaniem, wątki mogą być wykonywane równocześnie (współbieżnie). Równoczesny dostęp do wspólnych danych grozi utratą spójności danych, i w konsekwencji błędem działania programu.

Każdemu procesowi przydzielone zostają zasoby, takie jak:

- procesor
- pamięć
- dostęp do urządzeń wejścia-wyjścia
- pliki

Każdy proces posiada tzw. "rodzica". W ten sposób tworzy się swego rodzaju drzewo procesów. Proces może (ale nie musi) mieć swoje procesy potomne. Za zarządzanie procesami odpowiada Jądro systemu operacyjnego. Wykonanie musi przebiegać sekwencyjnie. Może przyjmować kilka stanów:

- działający,
- czekający na udostępnienie przez system operacyjny zasobów,
- przeznaczony do zniszczenia,
- właśnie tworzony itd.
- proces zombie

W skład procesu wchodzi:

- kod programu
- licznik rozkazów
- stos
- sekcja danych

Tworzenie procesów

- użytkownik za pomocą powłoki zleca uruchomienie programu, proces wywołujący wykonuje polecenie `fork`, lub jego pochodną
- System operacyjny tworzy przestrzeń adresową dla procesu, oraz strukturę opisującą nowy proces
- wypełnia strukturę opisującą proces
- kopiuje do przestrzeni adresowej procesu dane i kod, zawarte w pliku wykonywalnym
- ustawia stan procesu na działający
- dołącza nowy proces do kolejki procesów oczekujących na procesor
- zwraca sterowanie do powłoki użytkownika

Wykonywanie procesów

Dany proces rozpoczyna wykonywanie w momencie przełączenia przez Jądro systemu operacyjnego przestrzeni adresowej na przestrzeń adresową danego procesu oraz takie zaprogramowanie procesora, by wykonywał kod procesu. Wykonujący się proces może żądać pewnych zasobów, np. większej ilości pamięci. Zlecenia takie są na bieżąco realizowane przez system operacyjny.

Kończenie procesów

- proces wykonuje ostatnią instrukcję - zwraca do systemu operacyjnego kod zakończenia. Jeśli proces zakończył się poprawnie zwraca wartość 0, w przeciwnym wypadku zwraca wartość kodu błędu.
- w momencie zwrotu do systemu operacyjnego kodu zakończenia, system operacyjny ustawia stan procesu na przeznaczony do zniszczenia i rozpoczyna zwalnianie wszystkich zasobów, które w czasie działania procesu zostały temu procesowi przydzielone.
- system operacyjny po kolei kończy wszystkie procesy potomne w stosunku do procesu macierzystego.
- system operacyjny zwalnia przestrzeń adresową procesu. Jest to dosłowna śmierć procesu.

- *system operacyjny usuwa proces z kolejki procesów gotowych do uruchomienia i szereguje zadania. Jest to ostatnia czynność wykonywana na rzecz procesu.*
- *procesor zostaje przydzielony innemu procesowi.*

Sposób realizacji wszystkich tych działań jest różny dla różnych systemów operacyjnych.

Wywłaszczenie - technika używana w środowiskach wielozadaniowych, w której algorytm szeregujący (**scheduler**) może wstrzymać aktualnie wykonywane zadanie (np. proces lub wątek), aby umożliwić działanie innemu. Dzięki temu rozwiązaniu zawieszenie jednego procesu nie powoduje blokady całego systemu operacyjnego. W systemach bez wywłaszczenia zadania jawnie informują scheduler, w którym momencie chcą umożliwić przejście do innych zadań. Jeżeli nie zrobią tego w odpowiednim czasie, system zaczyna działać bardzo wolno. Oprócz tego wywłaszczenie umożliwia szczegółowe określenie czasu, w jakim dany proces może korzystać z procesora. Wywłaszczenie w niektórych systemach operacyjnych może dotyczyć nie tylko programów, ale także samego jądra – przykładem takiego systemu jest Linux.

Wywłaszczenie jest często ograniczane, na przykład procedury odpowiedzialne za obsługę przerw sprzętowych są zwykle niewywłaszczalne, co znacznie upraszcza ich konstrukcje ale wymusza też zadbanie o to, żeby szybko się kończyły umożliwiając działanie innym procesom. W systemie Linux przed wersją 2.6 niemożliwe było wywłaszczenie procesu który znajdował się w trybie jądra, co w pewnych sytuacjach mogło być powodem bardzo wolnej reakcji na działania użytkownika.

Popularne wielozadaniowe systemy operacyjne z możliwością wywłaszczenia procesów

- większość systemów uniksowych
- Microsoft Windows (95, ME, NT, 2000, XP, Vista, Windows 7)

Popularne wielozadaniowe systemy operacyjne bez możliwości wywłaszczenia procesów

- MS-DOS z nakładką Microsoft Windows 3.x i wcześniejsze
- Mac OS 9 i wcześniejsze

Wielowątkowość

Wielowątkowość - cecha systemu operacyjnego, dzięki której **w ramach jednego procesu może wykonywać kilka wątków** lub jednostek wykonawczych. Nowe wątki to kolejne ciągi instrukcji wykonywane oddzielnie. Wszystkie wątki tego samego procesu współdzielą kod programu i dane. W systemach nie obsługujących wielowątkowości pojęcia procesu i wątku utożsamiają się.

Systemy wielowątkowe to m.in. BeOS, Microsoft Windows 95, Windows NT i niektóre z rodziny Unix.

Cechy wielowątkowości

- wszystkie wątki wykonują się w ramach tylko jednego programu (procesu) - to znaczy, że wykonują ten sam kod wykonawczy (przykładowo: w systemach Unix/Linux każdemu wątkowi przydziela się konkretną funkcję do wykonania)
- wątki zostały wprowadzone aby zwiększyć wydajność w programowaniu współbieżnym, gdzie zachodzi potrzeba wykonania wielu zadań jednocześnie;
- wszystkie wątki procesu współdzielą tę samą wirtualną przestrzeń adresową (mają dostęp do tych samych "egzemplarzy" zmiennych, obiektów i struktur) i korzystają z tych samych zasobów systemowych;
- komunikacja między wątkami w odróżnieniu od procesów jest bardzo łatwa do wykonania - w przypadku wątków wystarczy odwoływać się do tych samych zmiennych i obiektów - komunikacja między procesami wymaga zastosowania mechanizmów IPC (ang. *InterProcess Communication*);
- współdzielenie wirtualnej przestrzeni adresowej niesie duże zagrożenie - jeden "wadliwy" wątek może zagrozić wykonaniu całego programu;
- każdy wielowątkowy system operacyjny zapewnia specyficzne metody synchronizacji wątków, które z wyżej wymienionych przyczyn należy bezwzględnie zaimplementować.

Typy obudów mikroprocesorów

Fizycznie mikroprocesor to krzemowa płytko o wielkości około 1 cm². Jest podatna na wpływ czynników zewnętrznych, stąd musi być umieszczona w ochronnej powłoce – ceramicznej, plastikowej lub metalowej. Obudowa ma wyprowadzenia (nóżki lub piny) pozwalające na montaż w płycie głównej.

Spotykane obecnie obudowy:

PGA – nóżki w kształcie symetrycznej siatki:

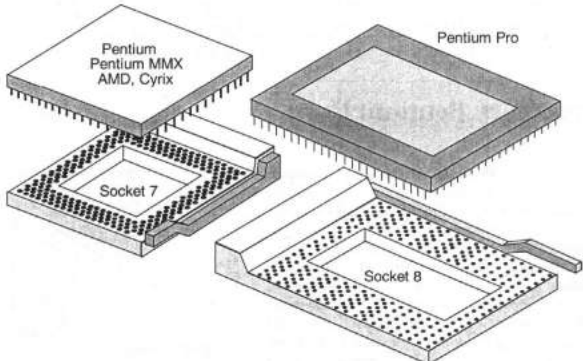
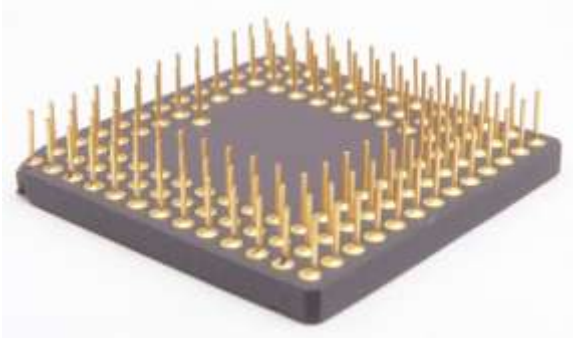
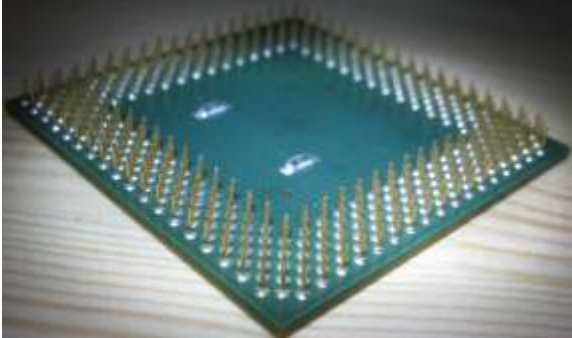
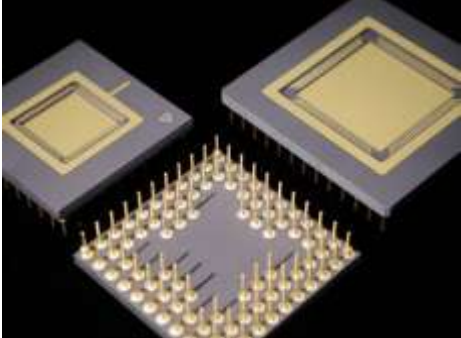


- PPGA – Plastic PGA – do osłony rdzenia wykorzystano plastikową powłokę,
- CPGA – Ceramic PGA – ceramiczna osłona rdzenia,
- FC-PGA – rdzeń przeniesiony na górną część obudowy w celu lepszego chłodzenia i zatopiony w plastiku,
- FC-PGA2 – podobnie jak w FC-PGA, ale dodatkowo rdzeń w plastikowej osłonie przykryto stalową blaszką.

SPGA – rozmieszczenie nóżek w rzędach i kolumnach jest niesymetryczne

SECC – specyficzny typ obudowy pochodzący z czasów kiedy nie potrafiono jeszcze umieścić pamięci Cache2 w strukturze rdzenia (Pentium II, III, Athlon). Mikroprocesor przylutowany był do płytki drukowanej wraz z pamięcią Cache L2, a całość w plastikowej obudowie w postaci kartridża.

SEPP – podobna do SECC z tą różnicą, że nie ma plastikowej osłony, stosowana w tańszych wersjach typu Celeron, Duron.

LGA – opracowany przez Intel, nóżki zastąpiono połączanymi stykami.

<p>PGA</p>  <p>Diagram illustrating the Pin Grid Array (PGA) package. It shows two processor types: Pentium, Pentium MMX, AMD, and Cyrix processors mounted on Socket 7, and Pentium Pro processors mounted on Socket 8. The processors are shown with their respective pins and sockets.</p>	<p>PGA</p>  <p>3D rendering of a PGA (Pin Grid Array) processor package, showing the dense array of pins on the underside of the chip.</p>
<p>PPGA</p>  <p>Photograph of a PPGA (Pin Grid Array) processor package, showing the dense array of pins on the underside of the chip.</p>	<p>CPGA</p>  <p>Photograph of CPGA (Copper Pin Grid Array) processor packages, showing the dense array of pins on the underside of the chip.</p>
<p>FC-PGA</p>  <p>Photograph of FC-PGA (Flip-Chip Pin Grid Array) processor packages. The top image shows an Intel processor on a green carrier with a label. The bottom image shows the processor mounted on a carrier with a dense array of pins.</p>	<p>SPGA</p>  <p>Photograph of an SPGA (Staggered Pin Grid Array) processor package, showing the dense array of pins on the underside of the chip. The chip is labeled with '55170580NC MALAY ES' and 'A802-18-15 SI 7-81 27'.</p>

SECC



SEPP



LGA

