

SYSTEMY OPERACYJNE

WYKŁAD PIERWSZY

JOANNA KOŁODZIEJCZYK

(JKOŁODZIEJCZYK@WI.ZUT.EDU.PL)

WIEDZA

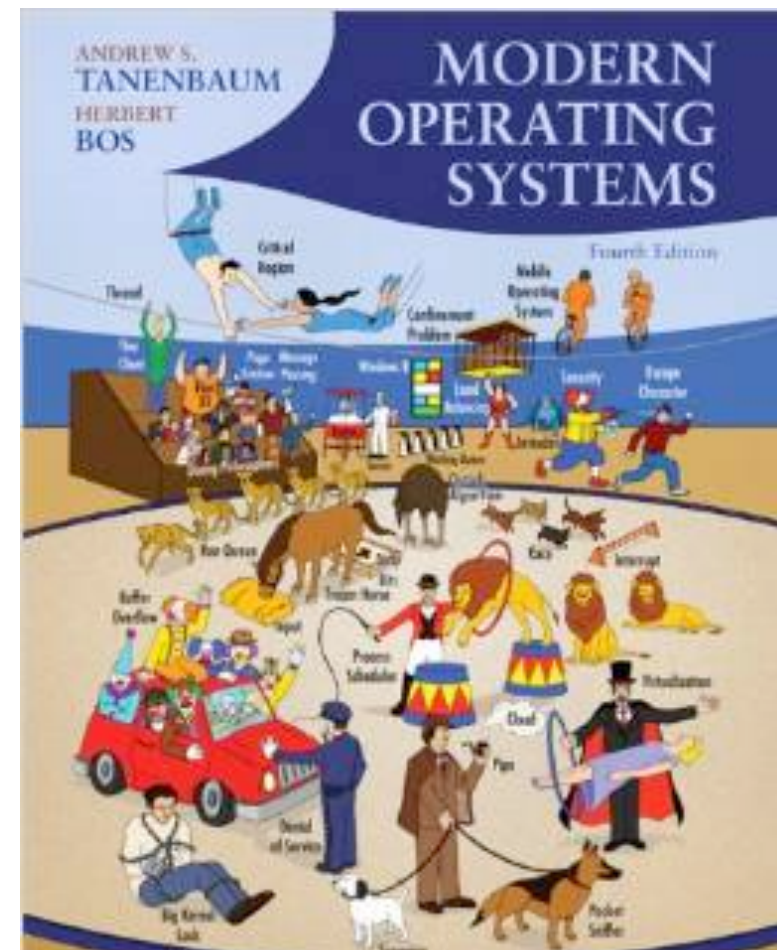
System operacyjny, historyczne rozwiązania, zastosowania, wady i zalety.

Budowa systemu, pojęcie jądra systemu, systemu plików, zastosowanie poszczególnych systemów plików oraz zna ich zalety i wady.

Pojęcia, takie jak pamięć wirtualna, proces, program, kolejowanie zadań, przetwarzanie w potoku, kompilacja i interpretacja, pojęcia systemu jedno i wielozadaniowego, a także wielodostępnego.

LITERATURA

- 1. Jasińska-Suwada A., Plichta St.: *Systemy operacyjne : przewodnik po laboratorium*, Politechnika Krakowska im. Tadeusza Kościuszki, Kraków 2010
- 2. Pytel K., Osetek S.: *Systemy operacyjne i sieci komputerowe. Cz. 1*, Wydawnictwa Szkolne i Pedagogiczne. Warszawa 2011
- 3. Tanenbaum A. S.: *Systemy operacyjne*, Helion 2012
- 4. Zdanowski S.: *Debian Linux. System operacyjny dla każdego. Pierwsze starcie*, Helion 2012
- 5. Ward B., *Jak działa Linux*, Wydawnicwo Helion, 2005



CEL KURSU



Zrozumienie koncepcji systemu operacyjnego



Jak działa system operacyjny, a co ważniejsze, dlaczego?



Jakie mechanizmy zapewnia SO

PROWADZĄCY

KONTAKT:

JOANNA.KOŁODZIEJCZYK@WSB.SZCZECIN.PL

JKOŁODZIEJCZYK@WI.ZUT.EDU.PL



MATERIAŁY DO ZAJĘĆ:
WIKIZMSI.ZUT.EDU.PL

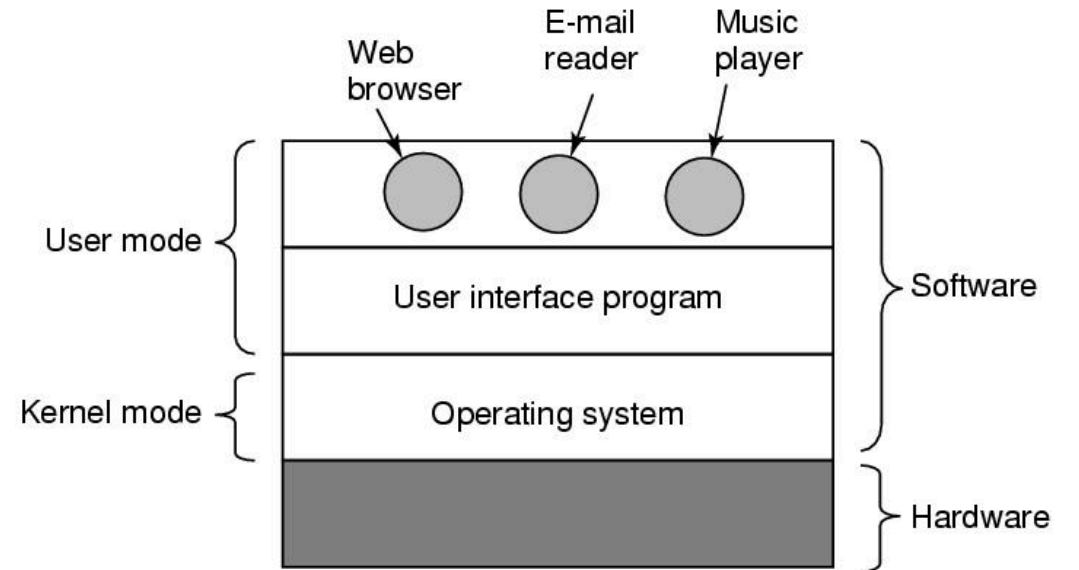
TREŚĆ KURSU



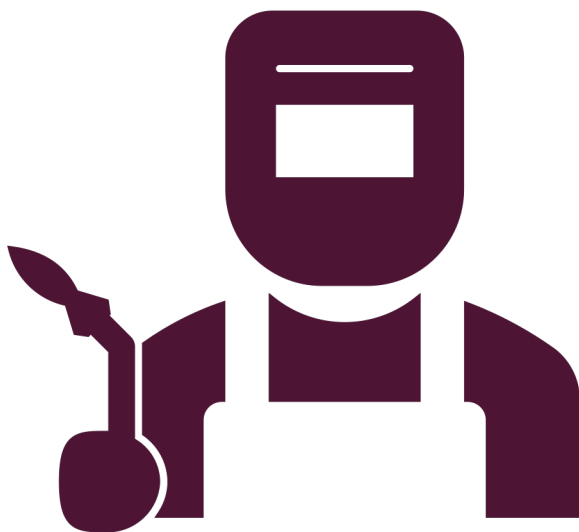
- Historia SO
- Abstrakcje zapewniane przez OS, mechanizmy oraz sposoby ich wykorzystania
- Procesy i zarządzanie nimi: synchronizacja i współbieżność
- Wątki
- Zarządzanie pamięcią, pamięć wirtualna
- Usługi SO: zarządzanie I/O - dyskami i systemy plików
- Systemy rozproszone

CO TO JEST SO?

- Jest programem, który działa (nieustannie) jako pośrednik pomiędzy użytkownikiem a sprzętem.
- Tworzy ułatwienia (abstrakcje), by łatwiej było używać sprzęt, by sprzęt nadzorować i nim zarządzać.



METAFORA



COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK

- OS jest jak manager
 - Zarządza zasobami operacyjnymi
 - Ustala zasady pracy (limituje zasoby, udostępnia tym którym mogą być udostępnione)
 - Rozwiązuje trudne i złożone zadania – sprowadza je do łatwiejszych

OS I HARDWARE (SPRZĘT)



COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK



Aplikacje



System operacyjny tworzy abstrakcję (model sprzętu) / kontroluje / pośredniczy w dostępie do zasobów sprzętowych (jakie zasoby?)



Obliczenia (CPUs) – złożone, wielordzeniowe



Pamięć ulotna (RAM) i pamięć trwała (dysk itp.)



Komunikacja (sieć, modem itp.)



Urządzenia wejścia / wyjścia (klawiatura, ekran, drukarka itp.)

KORZYŚCI DLA APLIKACJI

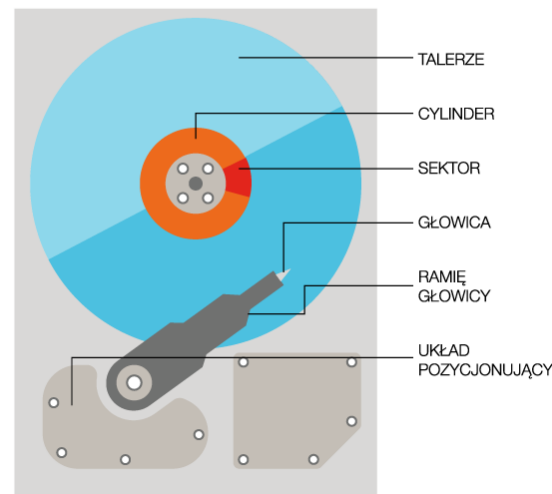
- Upraszcza złożony sprzęt dla aplikacji oraz dla programistów
- Urządzenie niezależne
 - wszystkie dyski wyglądają tak samo np. używają systemu plików
- Przenośny
 - ten sam program działa np. na rodzinie SO Windows 10
- Zarządza zasobami dla aplikacji
- Odizolowuje aplikacji i ich ochrona. Pomyśl ile aplikacji jest uruchomionych w tym czasie na Twoim telefonie lub komputerze.

PRZYKŁAD PORÓWNUJĄCY ŻYCIE Z / BEZ SYSTEMU OPERACYJNEGO

OS

```
file = open ("test.txt", O_WRONLY);  
write (file, "test", 4);  
close (file);
```

DYSK TWARDY



COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK

NO OS

- *Bloki, talerz, cylinder i sektor*
- *Gdzie jest ten plik na dysku? Który talerz, ścieżka i sektory?*
- *Kod musi się zmienić w innym systemie*

CO ROBI SYSTEM OPERACYJNY

- Zasoby (sprzętowe)
 - przydziela
 - chroni
 - odzyskuje
 - tworzy wirtualizacje

SO PRZYDZIAŁA ZASOBY (METAFORA)

- Skończone zasoby
 - Konkurencyjne wymagania
- Przykłady:
 - procesor
 - pamięć
 - dysk
 - sieć

- Rząd w kraju:
 - Ograniczony budżet
 - Obszar kraju
 - Zasoby naturalne

SO CHRONI ZASOBY (METAFORA)

- Ty nie możesz skrzywdzić mnie a ja nie mogę skrzywdzić Ciebie
- Pewien poziom bezpieczeństwa i ochrony

- Rząd w kraju:
 - Prawo, zasady, porządek

SO ODZYSKUJE (METAFORA)

- OS daje i OS zabiera
- Czasem jest to działanie mimowolne

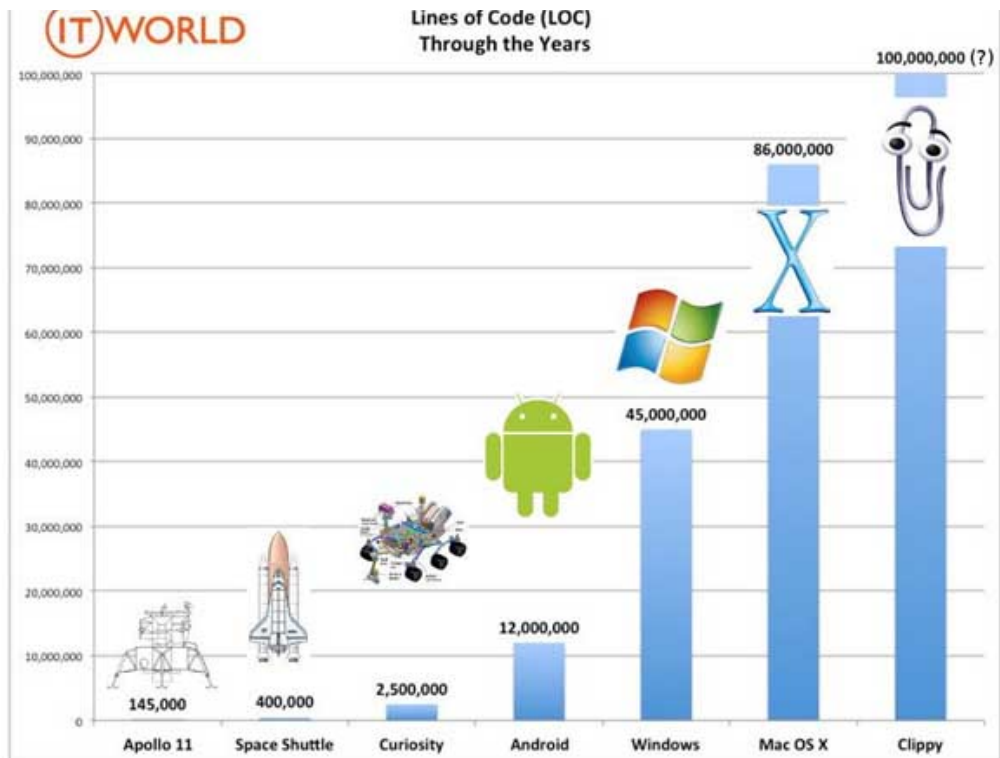
- Rząd w kraju:
 - Podatek dochodowy

SO WIRTUALIZUJE (METAFORA)

- Złudzenie nieskończonych prywatnych zasobów
- RAM vs. Dysk
- Podział czasu procesora

- Rząd w kraju:
- Opieka społeczna i ubezpieczenie

PO CO UCZYĆ SIĘ O OS?

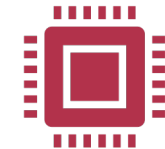


COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK



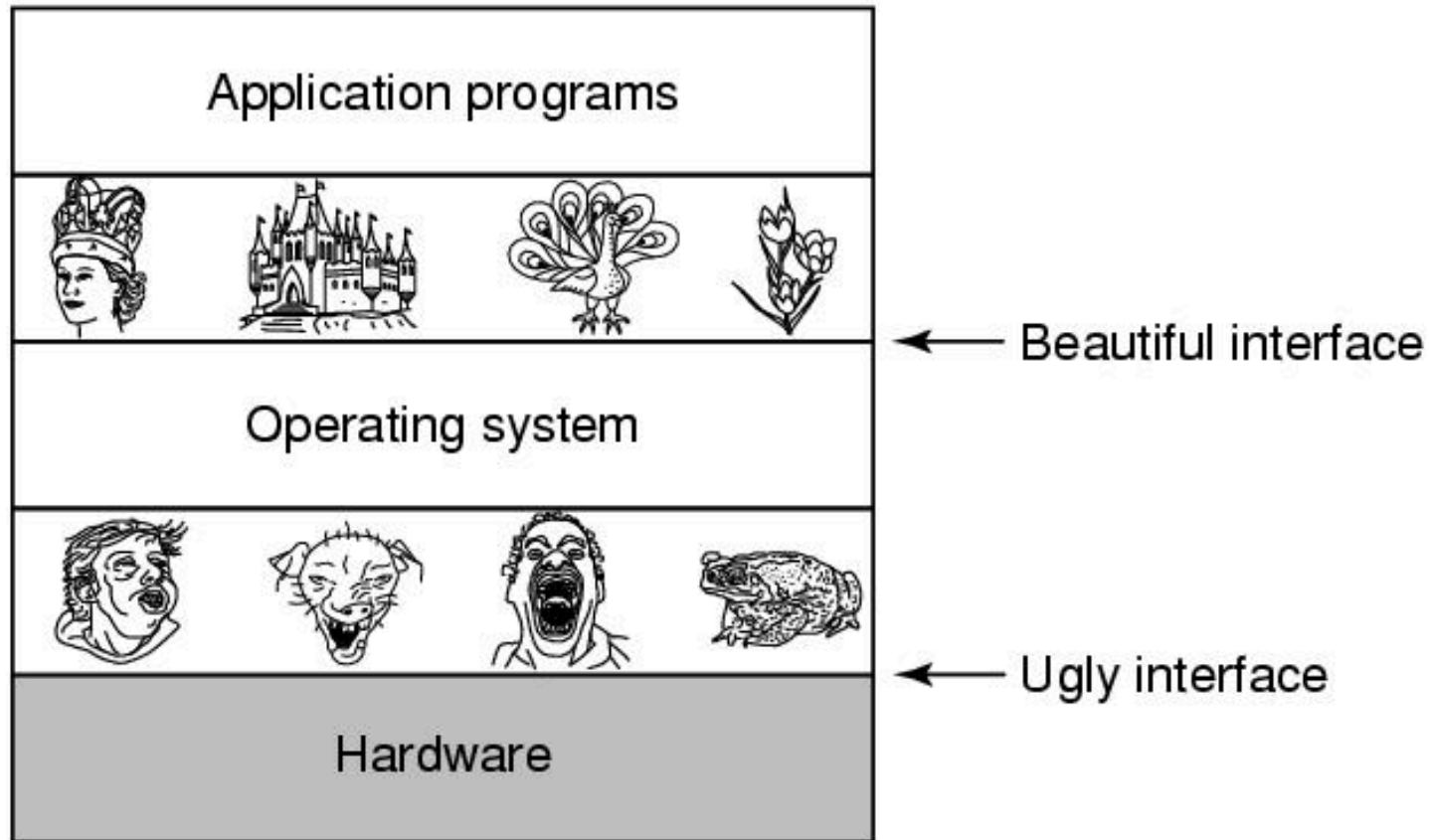
Podstawa dla innego oprogramowania

Bazy danych,
przeglądarki,
oprogramowanie komputerowe,.....



OS jest jednym z najtrudniejszych elementów oprogramowania do pisania i debugowania

Bezpośrednio rozmawia ze sprzętem
(bardzo brzydkie interfejsy)
Abstrakcja w czyste interfejsy
Są DUŻE



SYSTEMY
OPERACYJNE
ZAMIENIAJĄ BRZYDKI
SPRZĘT W PIĘKNY
ABSTRAKCJE.

QUIZ I

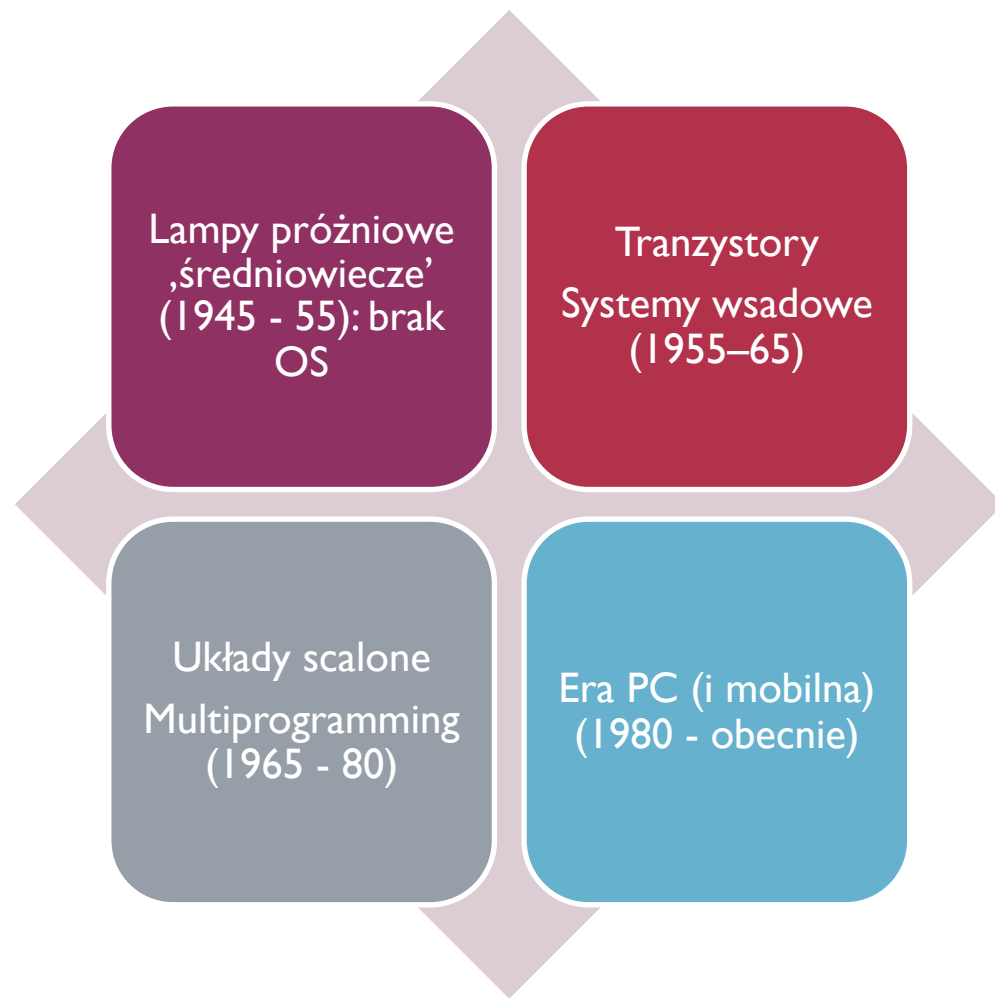


- Które z poniższych są komponentami SO?
 1. Edytor plików
 2. System plików
 3. Sterownik urządzenia
 4. Pamięć cache (podręczna)
 5. Przeglądarka internetowe
 6. Planer procesów

QUIZ 2



- Które z elementów są przykładami (A) abstrakcji i arbitrażu (B)?
 - A?B? Rozdzielanie pamięci pomiędzy wiele procesów
 - A?B? Wspieranie różnych rodzajów głośników
 - A?B? Niezmienny dostęp do dysku twardego czy SSD



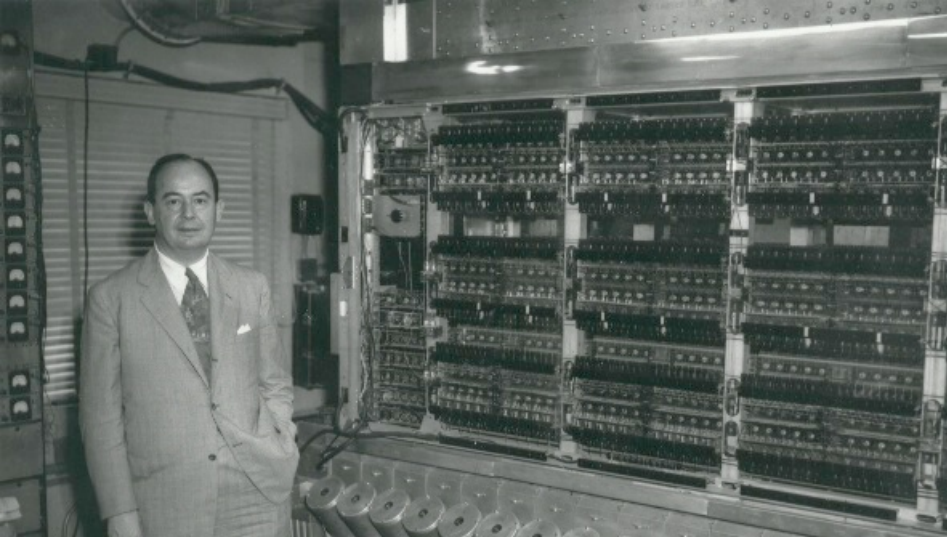
HISTORIA SYSTEMÓW OPERACYJNYCH

DARK AGE



COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK

- Komputery zbudowane z lamp próżniowych,
- ENIAC (*Electronic Numerical Integrator and Computer*) – budowany w latach (1943-1946), masa: 30 ton, 5000 operacji dodawania na sekundę, programowanie przez przełączniki i przełączanie kabli.
- Zaprojektowała i wykorzystała je jedna grupa ludzi je
 - Ochrona i wirtualizacja nie są potrzebne!
- Jak przydzielić i odzyskać czas obliczeń?
 - Arkusz rejestracyjny na ścianie!



VON NEUMAN

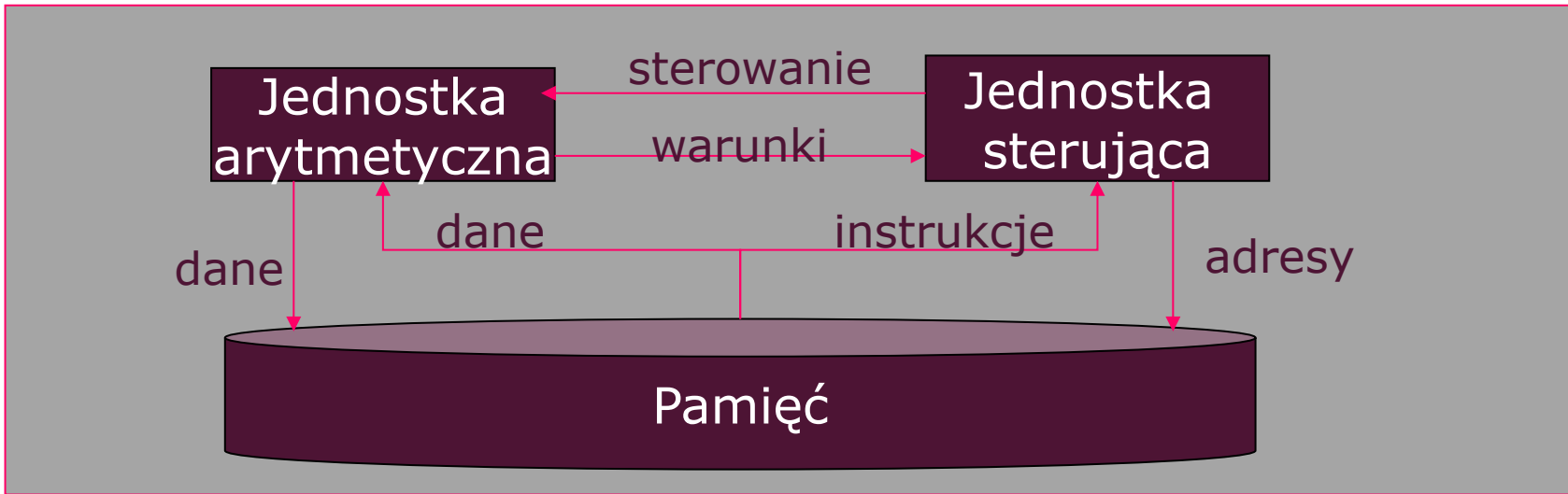
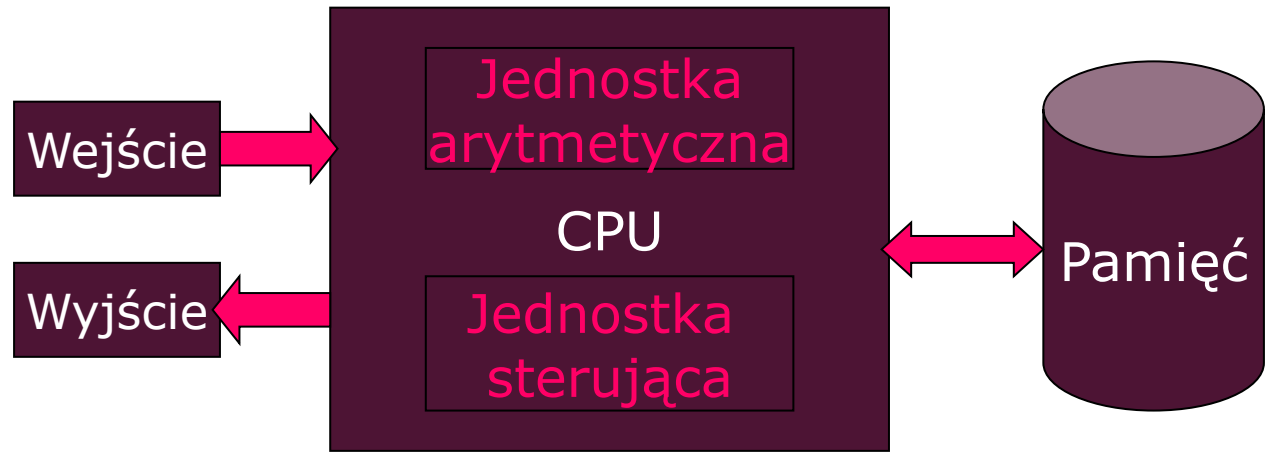
- IBM 701 (1953) – do badań naukowych.
- EDVAC (1952) – zbudowany w oparciu o architekturę von Neumana.

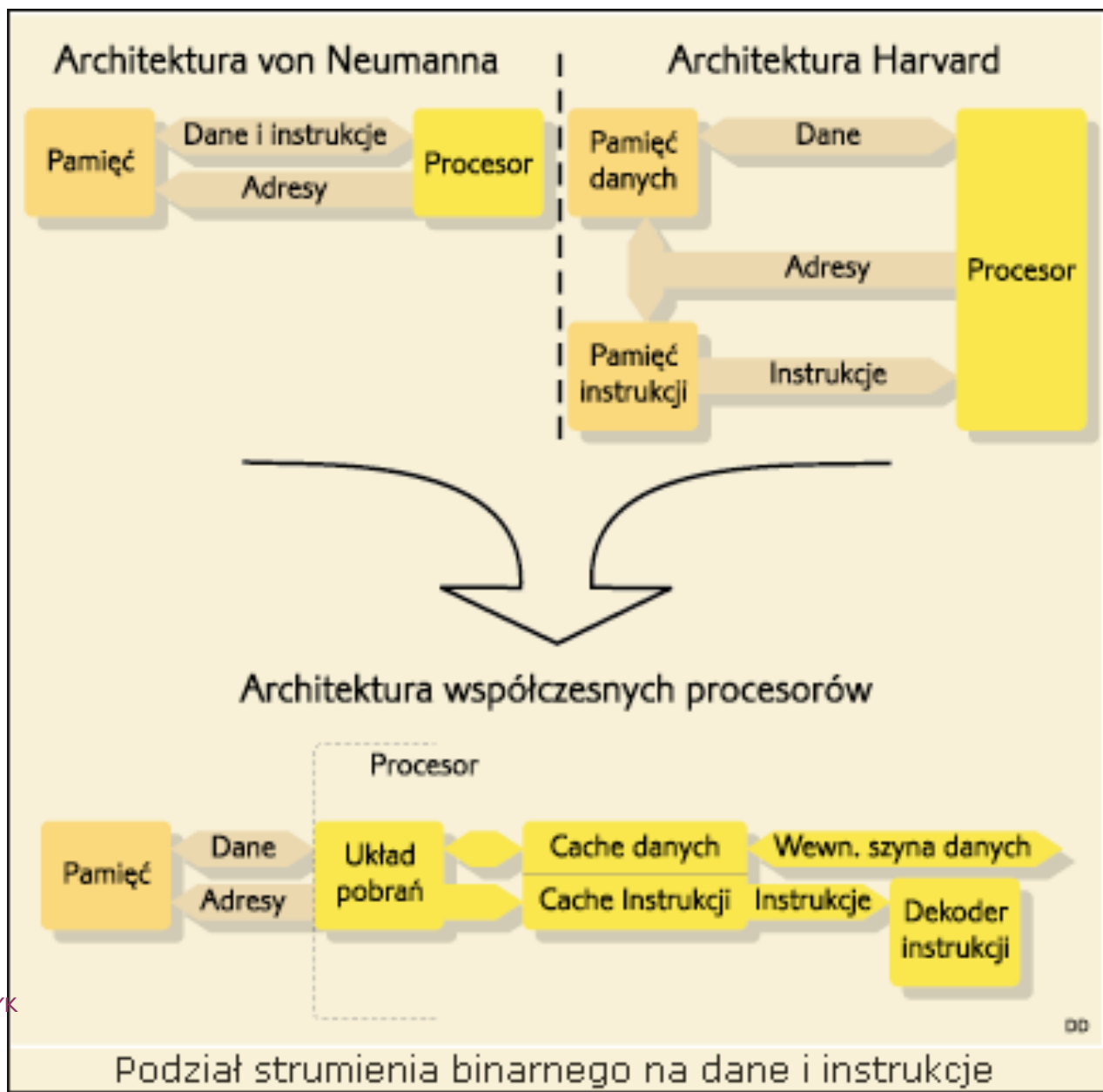
COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK



ARCHITEKTURA VON NEUMANN'A

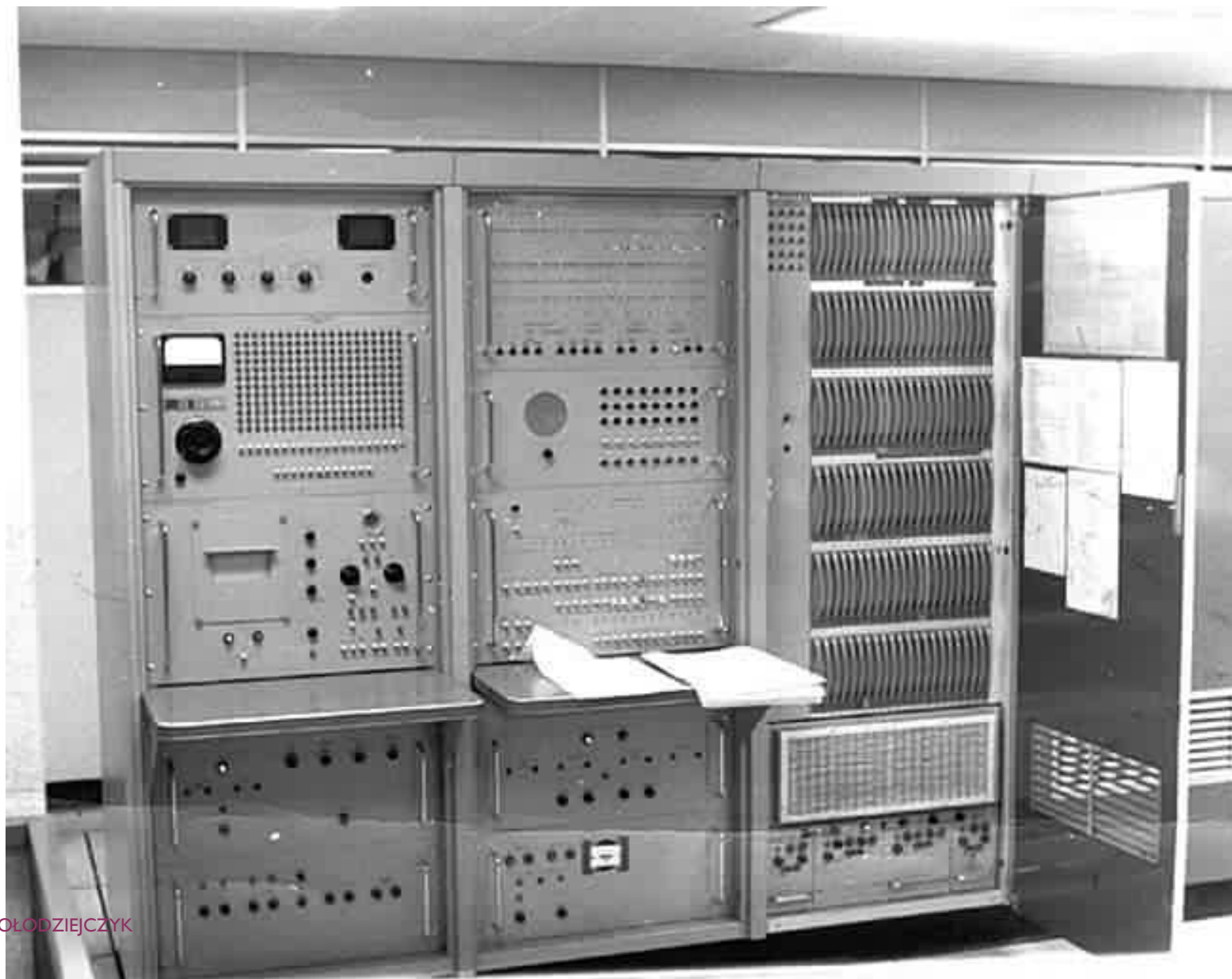
- Komputer powinien posiadać:
 - **Procesor** - arytmometr, który wykonuje działania arytmetyczne, logiczne i inne.
 - **Pamięć** złożoną z elementów przyjmujących stany 0 lub 1.
 - Możliwość **wprowadzania** danych i **wyprowadzania** wyników oraz sterowania.
- Działanie komputera związane jest z realizacją programu i obróbką danych zakodowanych w pamięci.
- Dane i instrukcje przechowywane są w pamięci.
- Stan maszyny i wejść sterowany programowo.
- Program może zawierać rozkazy warunkowe, które umożliwiają rozgałęzienia i skoki, może się także modyfikować podczas wykonywania.





- UPO → użytkownik=programista=operator. Komputer obsługiwany za pomocą konsoli operatorskiej.
- Zadania UPO:
 1. Pisze program.
 2. Rozkazy wprowadza do pamięci ręcznie ustawiając przełączniki na płycie czołowej (później za pomocą taśmy lub karty perforowanej). Ustawia adres startowy i uruchamia program.
 3. Debugowanie (kontrola poprawności programu): obserwacja lampek na konsoli (odczytywanie komórek pamięci), poprawki wprowadzane z konsoli.
 4. Odbiera rezultaty drukowane na taśmie papierowej lub dziurkowane na karcie perforowanej.
- Harmonogram pracy – koniecznie przestrzegany.

JAK OBSŁUGIWANO KOMPUTERY PIERWSZEJ GENERACJI?



COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK



COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK

CZY WIDAĆ SŁABE PUNKTY?

Określony czas pracy:

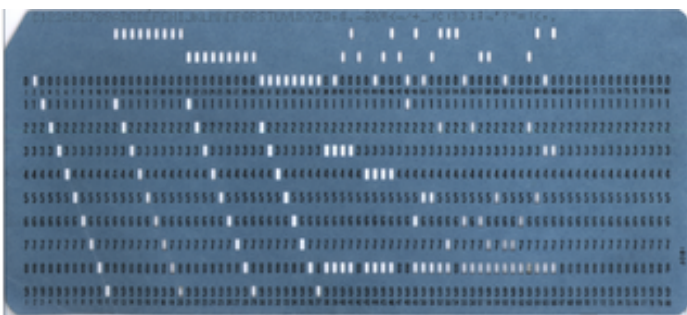
- założony z góry czas pracy może okazać się za krótki na poprawienie błędów,
- ktoś kończy zbyt wcześnie i czas komputera jest niewykorzystany.

Programowanie w języku maszynowym.

Wszystkie stałe fragmenty (np.. obsługi urządzeń) realizowane od podstaw.

Błędy wymuszają zaczynanie wszystkich kroków obliczeń od początku.

TRANZYSTORY I SYSTEMY WSADOWE (1955-65)



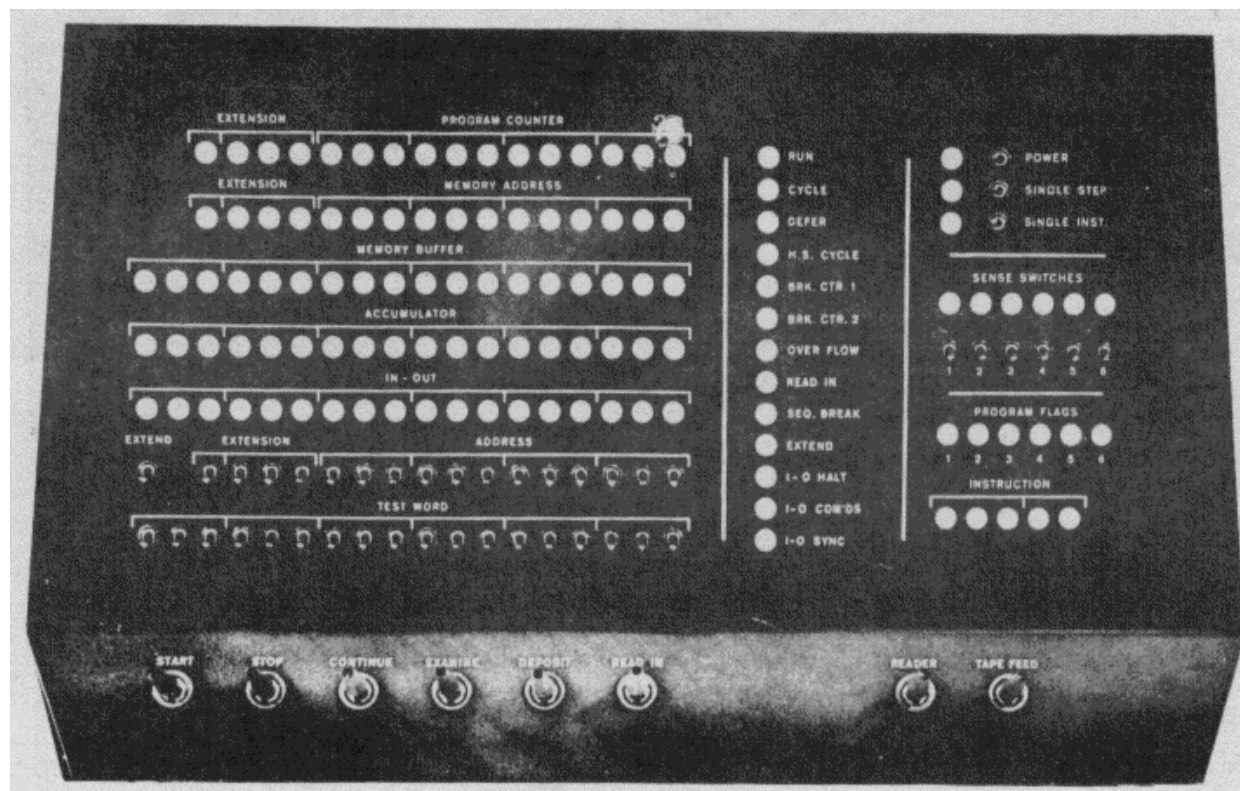
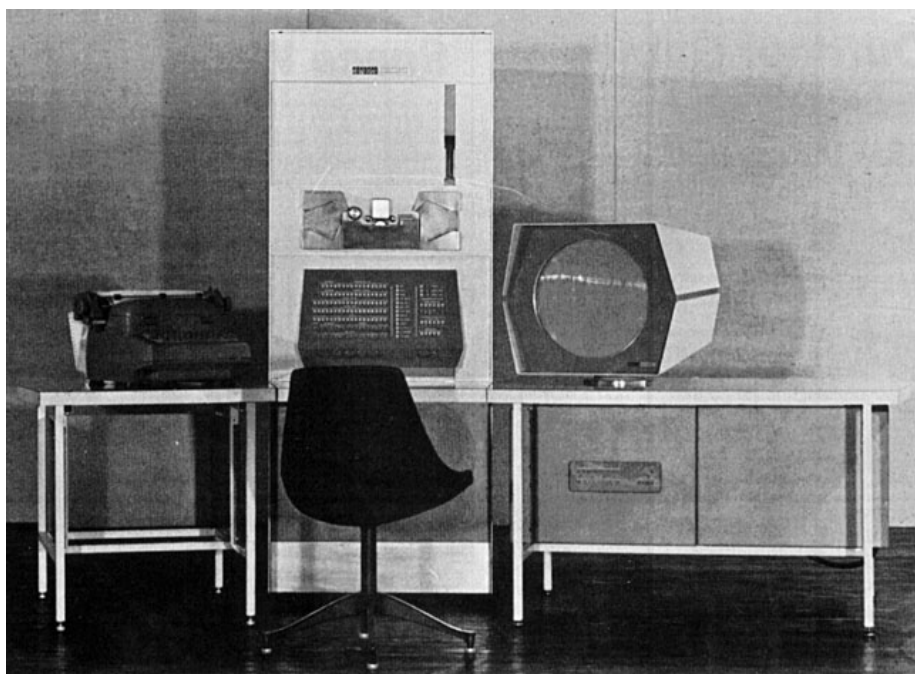
COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCYK

- Tło:
 - W tym czasie komputery są używane tylko do „obliczania” (zamiast rozrywki itp.)
 - Użytkownicy piszą program za pomocą „karty dziurkowanej”
 - Wdziurkuj swój program w kartę
 - Przynieś karty do operatorów komputerów
 - Wróć po dniu, aby uzyskać wynik

HISTORIA KOMPUTERÓW – ETAP 2

- Druga generacja (1955-1965)
 - Komputery zbudowane z tranzystorów.
 - Mniej się grzały i były znacznie tańsze.
 - **PDP-1** - DEC (Digital Equipment Corporation) – firma założona w 1957.
 - **IBM 1401** do wprowadzania i wyprowadzania danych.
 - **IBM 7094** – do obliczeń numerycznych.

PDP-1



CO NOWEGO W OBSŁUDZE?

- Wprowadzono programowanie:
 - języki: FORTRAN, ALGOL,
 - tworzone programy:
 - ładujące,
 - łączące,
 - biblioteki funkcji,
 - kompilatory,
 - podprogramy wykonujące operacje wejścia-wyjścia (zawierały wiedzę o tym jaki używać bufory urządzeń, znaczników, rejestrów (dołączane z biblioteki)).

PODZIAŁ RÓL



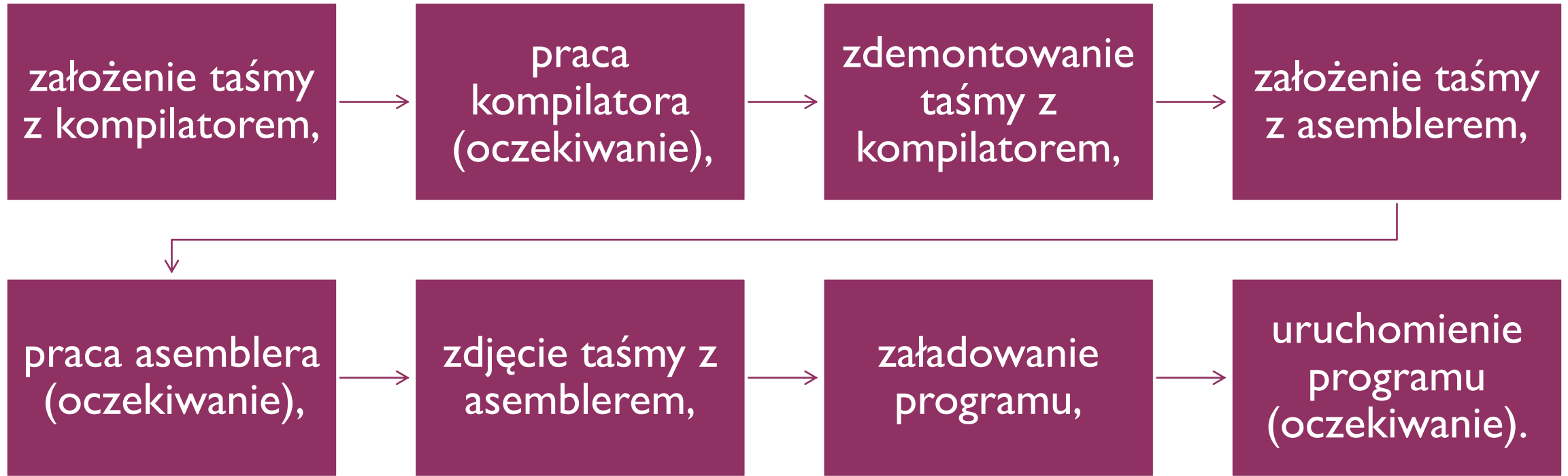
UŻYTKOWNIK NIE JEST JUŻ
OPERATOREM.



UŻYTKOWNIK PISZE
PROGRAMY.



OPERATOR WYKONUJE PRACĘ
SYSTEMU OPERACYJNEGO.



PRACA OPERATORA

TRYB WSADOWY

Grupowanie zadań o podobnych wymaganiach.

Schemat pracy:

- programista daje program operatorowi,
- operator przygotowuje wsad,
- wykonywanie grup programów,
- wyniki wędrują do programistów,
- jeżeli błąd w programie, to operator wyprowadza zawartość pamięci i rejestrów i przekazuje programiście, który na tej podstawie poprawia błędy.

MONITOR REZYDUJĄCY - OS

Monitor - automatyczna procedura przejść pomiędzy zadaniami.

Rezydentny – umieszczony w pamięci operacyjnej komputera przez cały czas pracy.

Jak działa monitor rezydujący?:

uruchamiany zaraz po włączeniu komputera,

gdy zlecono program do wykonania, przekazuje sterowanie do programu,

program po wykonaniu zwraca sterowanie do monitora, który wyznacza kolejny program do uruchomienia.

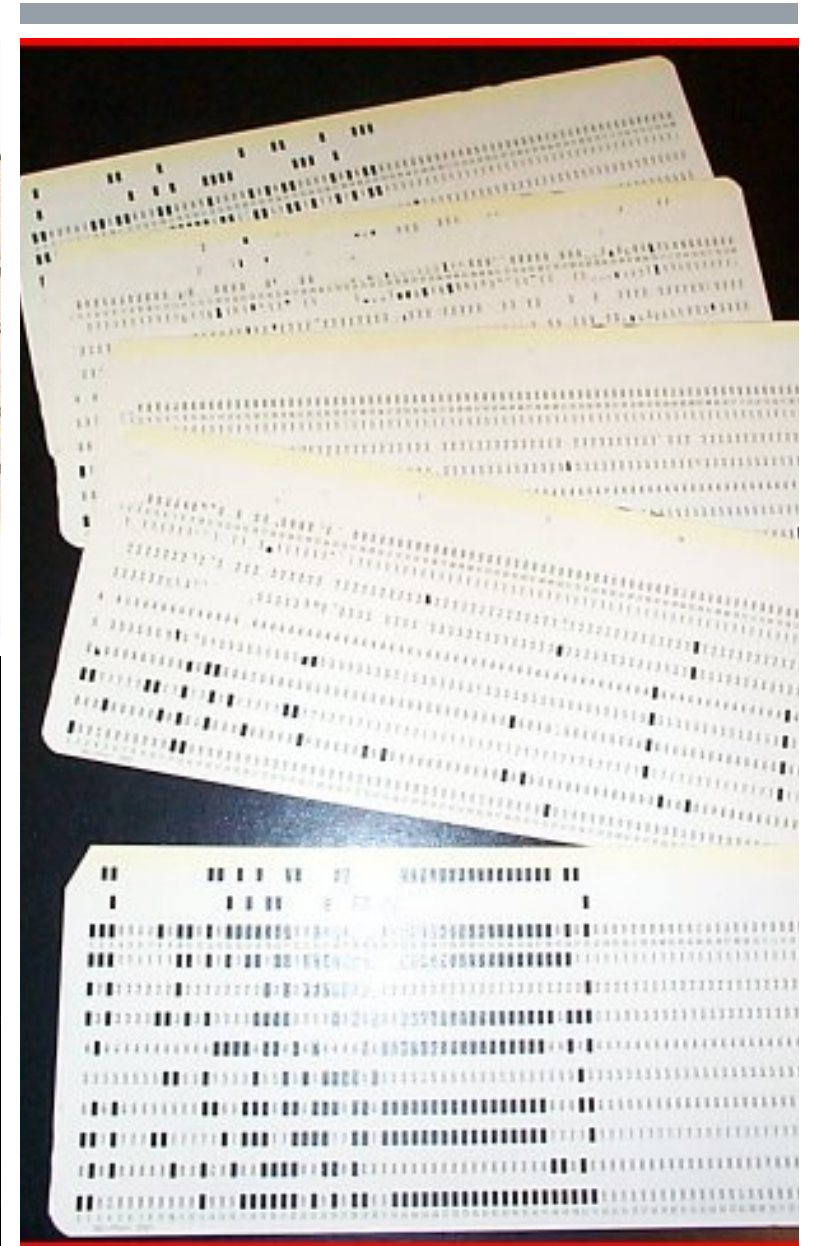
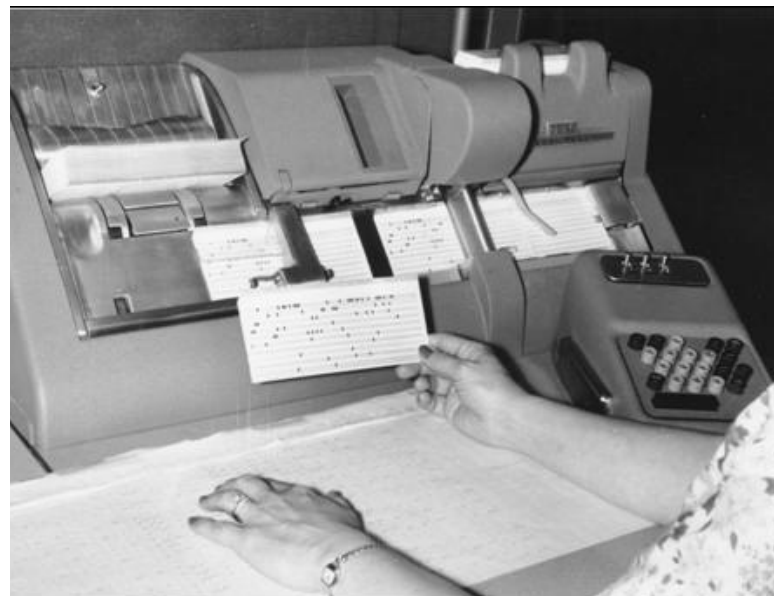
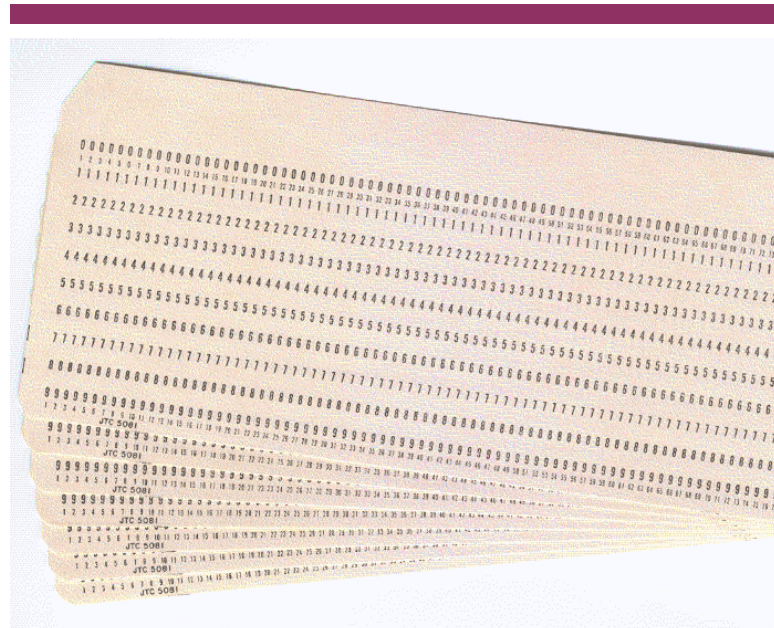
PRZYKŁAD DZIAŁANIA

- Posługujemy się kartami.
- Część to karty sterujące, które zawierają informację (rozkaz) jaki program wykonać i skąd wziąć dane.
- Części monitora (systemu operacyjnego):
 - interpreter kart sterujących: czytanie i wykonanie poleceń z kart podczas wykonywania zadania, wywołanie programu ładującego,
 - program ładujący: wprowadza do pamięci programy systemowe lub użytkowe,
 - programy obsługi urządzeń wejścia-wyjścia.

KARTY PERFOROWANE

- <https://www.vintageisthenewold.com/living-like-the-pioneers-code-fortran-in-a-punch-card/>

COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK



PRZYKŁAD DZIAŁANIA MONITORA

\$job – początek zadania (użytkownik, konto, priorytet)

\$ftn – kompilator Fortranu

program zapisany w języku Fortran

\$load

\$run – program użytkownika

dane do programu

\$end – koniec zadania

\$job – początek zadania (użytkownik, konto, priorytet)

(\$) karty sterujące

ULEPSZENIE OBSŁUGI I/O

- Wprowadzenie trybu pośredniego oraz wykorzystanie szybszych taśm magnetycznych.
 1. tryb bezpośredni:
 - czytnik kart → jednostka centralna → drukarka wierszowa
 2. tryb pośredni:
 - czytnik kart → taśma magnetyczna → jednostka centralna → taśma magnetyczna → drukarka wierszowa

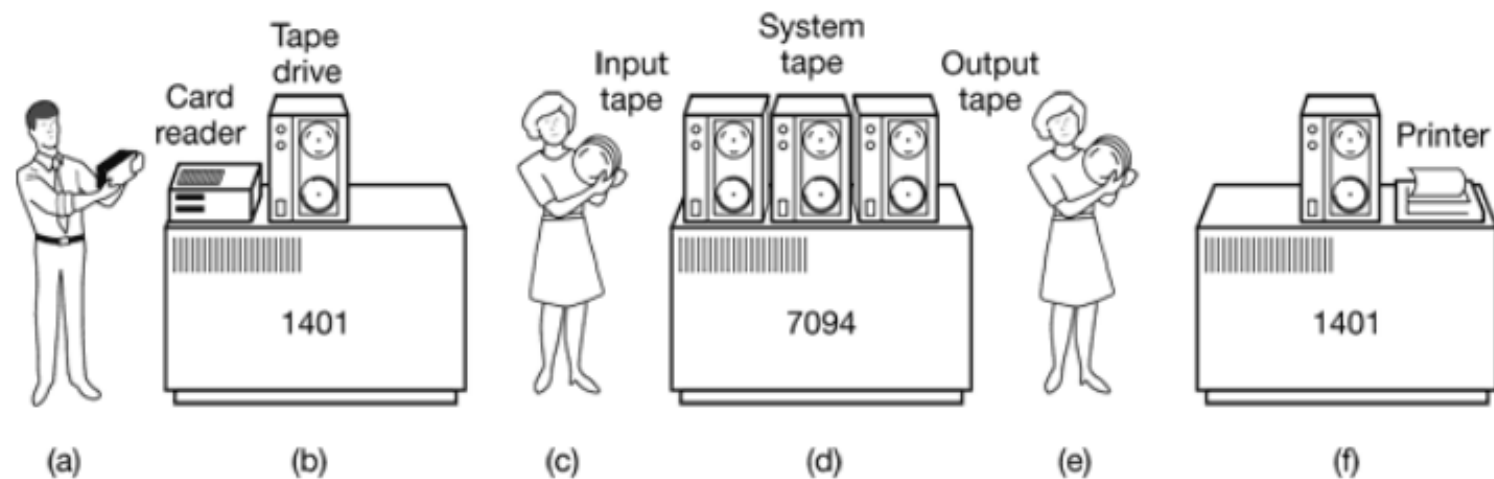
ZALETY TRYBU POŚREDNIEGO



COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK

1. Nie trzeba było wprowadzać zmian w programach – przy np.. odwołaniu do czytnika karty wywołana była ta sama procedura systemowa.
2. Na jeden procesor mogła przypadać duża liczba zestawów czytnik-
taśma i taśma-drukarka – wynikiem czego było przyspieszenie
pracy.

DANE PRZECHOWYWANE NA TAŚMACH MAGNETYCZNYCH TRYB WSADOWY



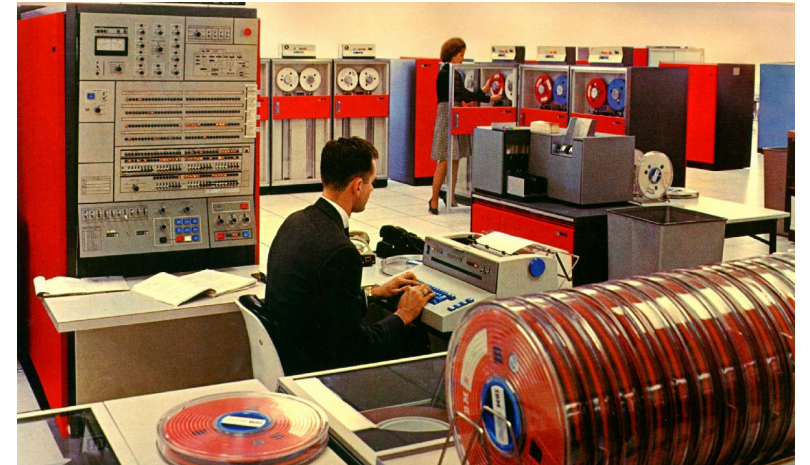
MULTIPROGRAMMING (1965-80)

- Problemy z systemami wsadowymi?
 - Interaktywność
 - Czy chcesz poczekać dzień, aby dowiedzieć się, że Twój program się nie kompiluje?
 - Wielu użytkowników nie może jednocześnie uzyskać dostępu do komputera
 - Wydajność
 - Procesor jest bezczynny, gdy komputer wykonuje operacje we / wy
- Wieloprogramowanie
 - Wiele zadań jest wykonywanych w tym samym czasie, tak jakby były wykonywane jednocześnie
 - Teraz wielu użytkowników może korzystać z tej samej maszyny jednocześnie

HISTORIA KOMPUTERÓW – TRZECIA GENERACJA

- Układy scalone.
- Procesory potokowe.
- PDP-11 – wyposażony w pamięć cache 1970,
- IBM 360/370 (1968)
- CDC 6600

COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK



CO NOWEGO W OPROGRAMOWANIU?

Język programowania C.

1972 – powstał system operacyjny UNIX
napisany w języku C.

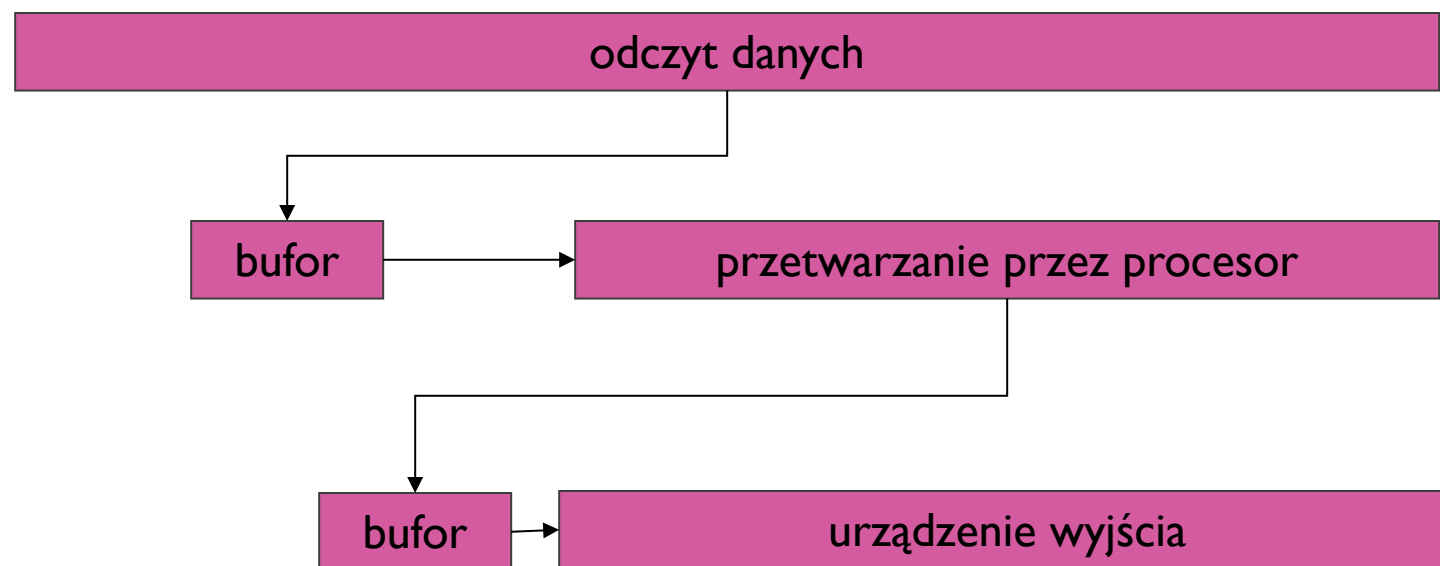
Buforowanie.

Spooling

Stosowanie wieloprogramowości.

BUFOROWANIE

- **Jednoczesne** wykonanie obliczeń i operacji wejścia-wyjścia dla jednego zadania.



SYSTEMY DYSKOWE

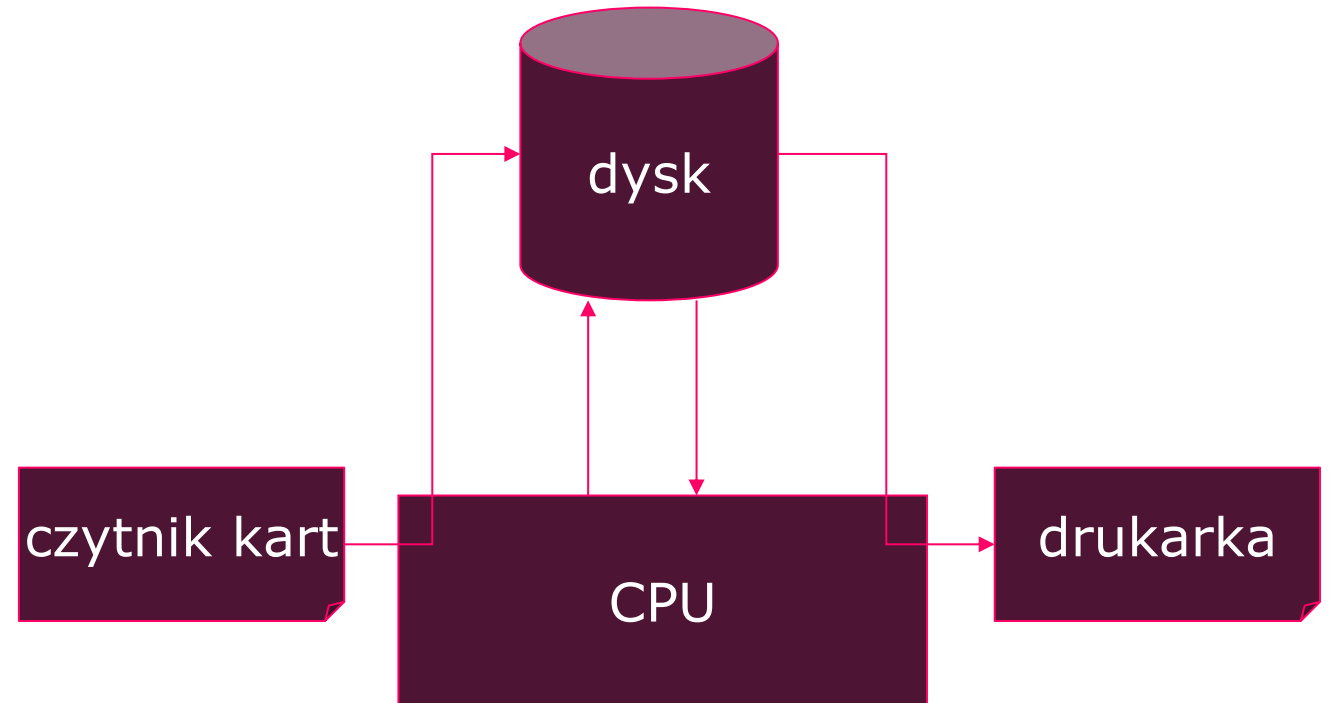


COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK

- Pierwszy dysk magnetyczny powstały w 1956 (IBM), pojemność: 5 MB.
- W dysku głowica ma dostęp do dowolnych obszarów (dostęp bezpośredni). Taśmę trzeba było przewijać.
- Zawartość z czytnika kart mogła być przesyłana bezpośrednio na dysk.
- Obraz rozmieszczenia kart na dysku przechowywany był przez SO w tablicy.
- Wyniki przechowywane były na dysku, aż do momentu wydrukowania.

SPOOLING

- Równoczesne wykonanie operacji wejścia-wyjścia i obliczeń dla **innych** zadań.
- Dysk staje się wielkim buforem do czytania z urządzeń wejściowych z maksymalnym wyprzedzeniem i przechowywania danych wyjściowych.
- **Spooling** wytwarza **pulę zadań**. Przeczytane wcześniej zadania czekają na wykonanie, a system operacyjny wybiera je tak, by jak najlepiej wykorzystać czas CPU.



MULTiplexed Information and Computing Service

- Zainicjowany przez MIT, Bell Labs i General Electric

Zaprojektowany, aby obsługiwać setki użytkowników na maszynie o wiele mniejszej mocy niż iPhone 5

- Użytkownicy umieli pisać małe, wydajne

Sukces techniczny, ale nie tak komercyjny

- Bell Labs i GE wycofali się, zanim SO został wydany

MULTICS

UNIX

Napisane przez Kena Thompsona i Dennisa Ritchie z Bell Labs na PDP-11 (1971)

Projekt rozpoczął się, ponieważ Ken chciał zagrać w grę „Space Travel”

Pierwotnie nazwany „Unics” przez Briana Kernighana

W 1973 r. Ritchie wynalazł język programowania C, aby ułatwić rozwój Uniksa

WIELOPROGRAMOWOŚĆ PODZIAŁ CZASU

Z chwilą przerwania zadania (np.. oczekiwanie na zakończenie operacji wejścia-wyjścia) przechodzi się do wykonania kolejnego zadania z puli oczekujących.

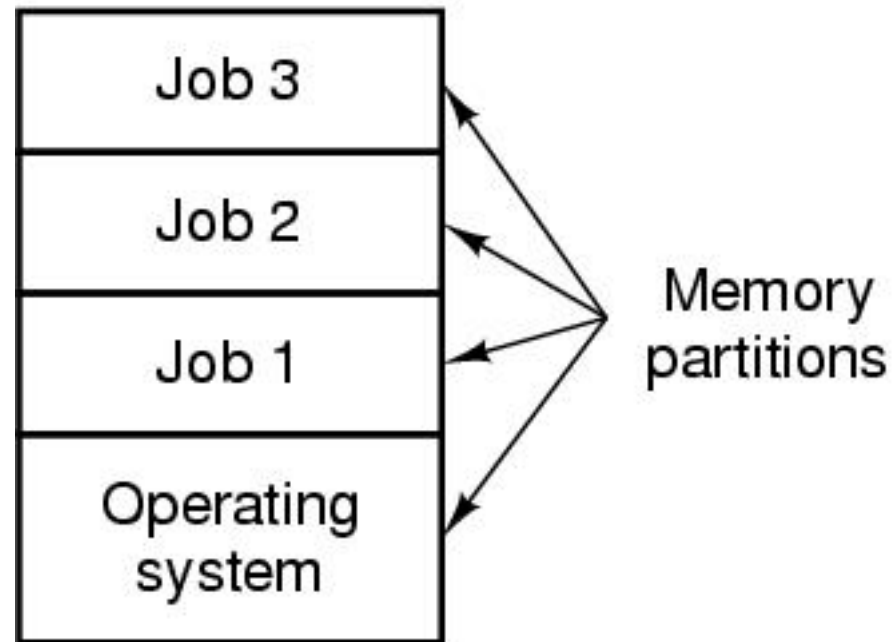
Zadanie przerwane wraca do puli zadań do wykonania po otrzymaniu oczekiwanych danych.

Wszystkie zadania muszą być przechowywane w pamięci.

Niezbędne jest planowanie przydziału procesora dla zadań oczekujących i wykluczanie konfliktów pomiędzy zadaniami.

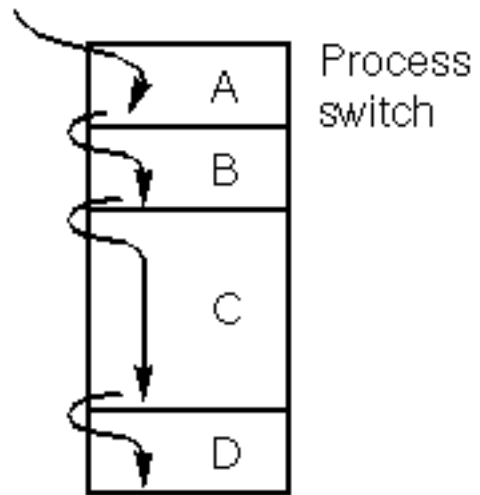
Zadania mogą pochodzić od różnych użytkowników (wielodostęp).

SYSTEM WIELOPROGRAMOWY Z TRZEMA ZADANIAMI W PAMIĘCI.

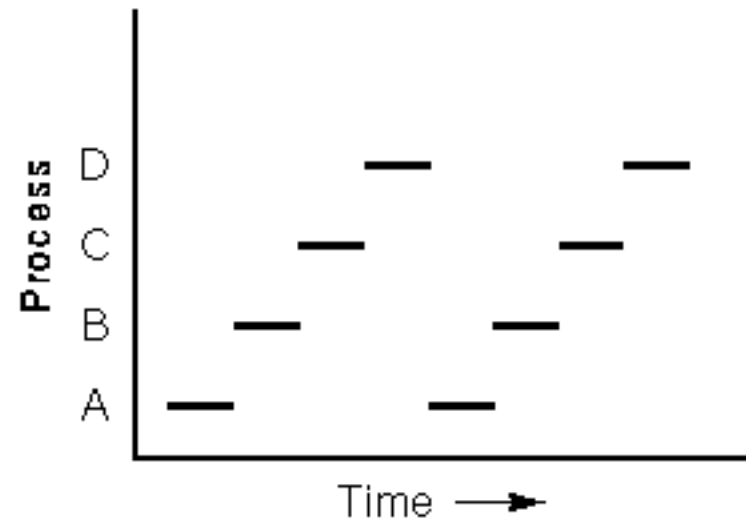
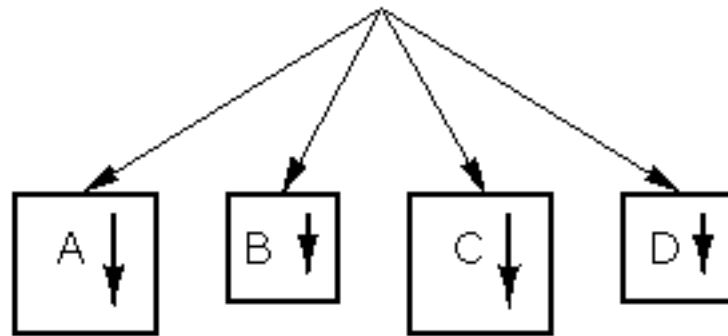


PRZYKŁAD

One program counter



Four program counters



SYSTEM STAJE SIĘ INTERAKCYJNY



Zapewnienie komunikacji on-line
użytkownika i systemu.



Użytkownik wydaje rozkazy
bezpośrednio do systemu i od razu
uzyskuje odpowiedzi.



Wprowadzenie interakcyjnych
edytorów, programów było zasługą
łatwiejszego programowania.

SYSTEM PLIKÓW



Plik – zestaw informacji zdefiniowany przez jego twórcę (programy, dane) lub ciąg bitów, bajtów, wierszy, rekordów.



Zarządzanie pamięciami masowymi urzeczywistnia abstrakcyjne pojęcie pliku.



Pliki organizuje się w grupy – katalogi.



Ze względu na możliwość istnienia wielu użytkowników nadzorowany jest dostęp do plików.

PODSUMOWANIE

- W trzeciej generacji komputerów powstały systemy operacyjne wielozadaniowe z podziałem czasu. Ich cechy to:
 - system jest interakcyjny,
 - posiada planowanie przydziału procesora,
 - zawiera ochronę pamięci,
 - nadzoruje pamięci dyskowe,
 - możliwe jest by wielu użytkowników dzieliło jeden komputer.

IBM wprowadza komputer PC w 1981 roku, wykorzystując procesory Intel

System operacyjny na każdym komputerze

W tym czasie system operacyjny o nazwie CP / M już pracuje na procesorach Intel

Bill Gates początkowo zaproponował IBM zawarcie umowy na CP / M

Założyciel CP / M odmówił spotkania z IBM

IBM wrócił do Billa Gatesa i kupił system operacyjny o nazwie DOS (Disk Operating System), zmodyfikował go i zmienił jego nazwę na MS-DOS

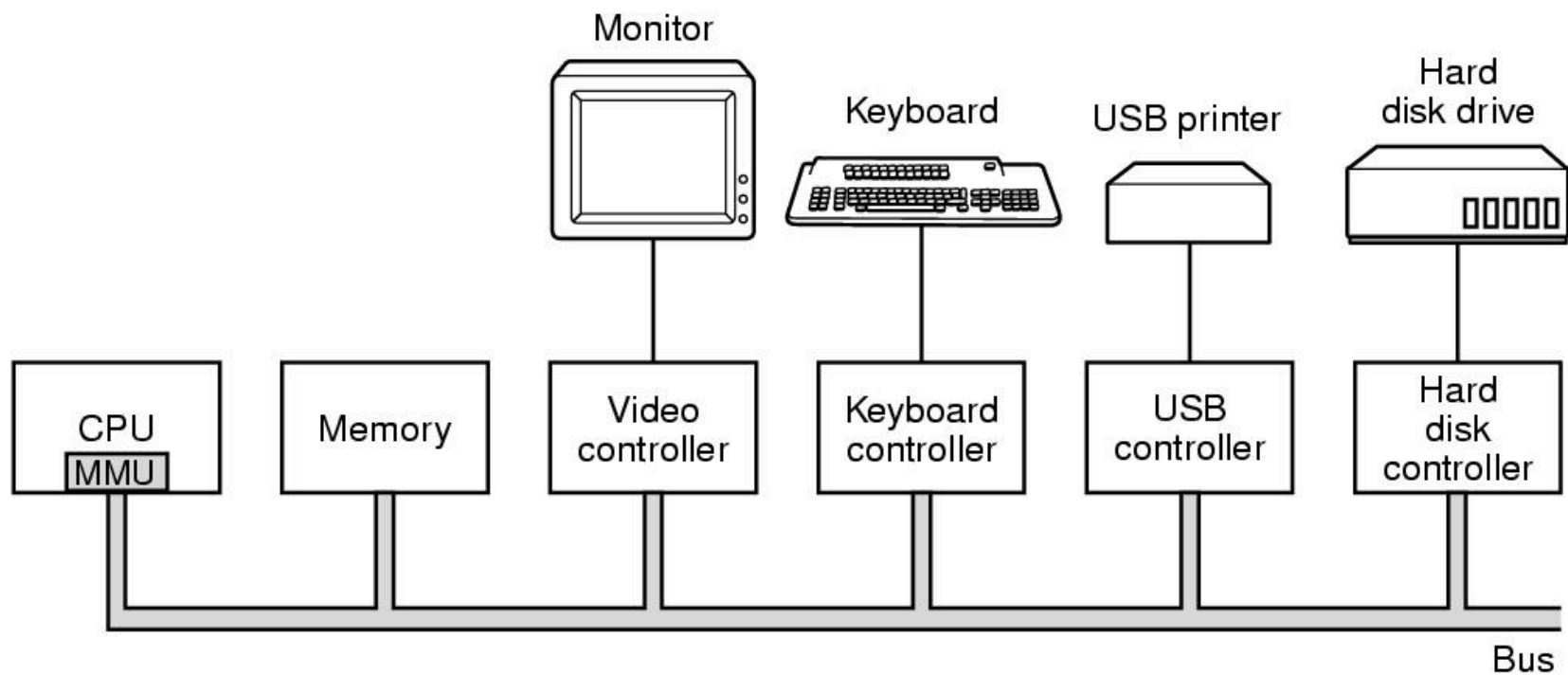
ERA PC (1980- WSPÓŁCZEŚNIE)

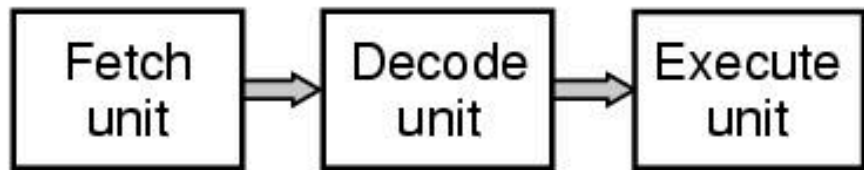
CZWARTA GENERACJA KOMPUTERÓW

- Układy scalone o bardzo dużej skali integracji.
- Maszyny wieloprocessorowe.
- Superkomputery wektorowe.
- IBM PC
- VAX9000

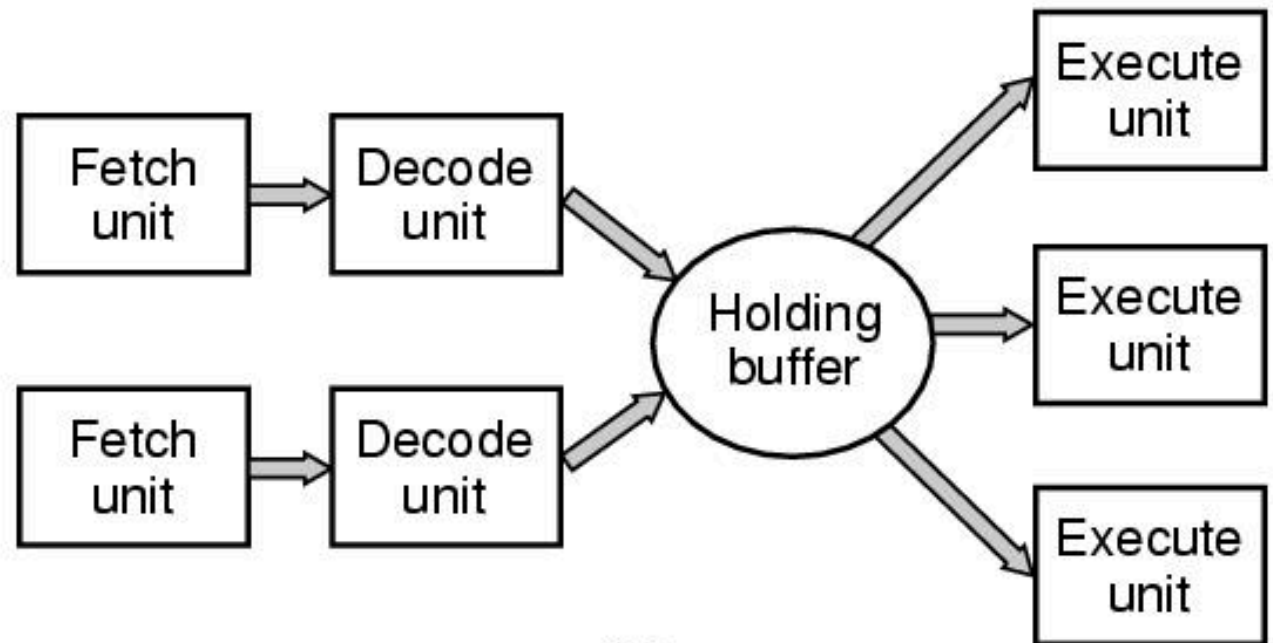


NIEKTÓRE KOMPONENTY PROSTEGO KOMPUTERA OSOBISTEGO





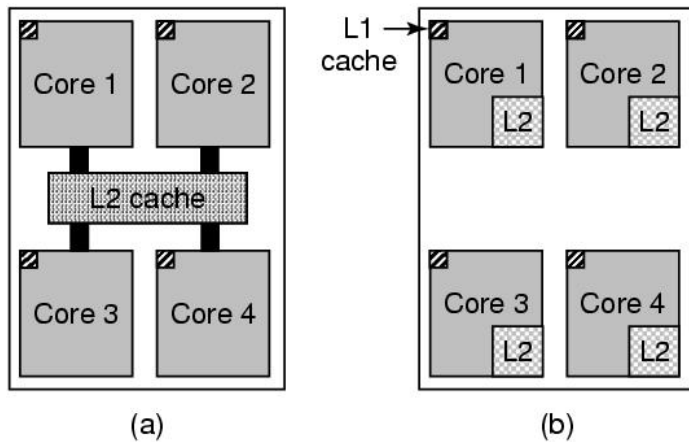
(a)



(b)

(A) TRÓJSTOPNIOWY PIPELINE CPU (B) SUPERSKALARNY PROCESOR

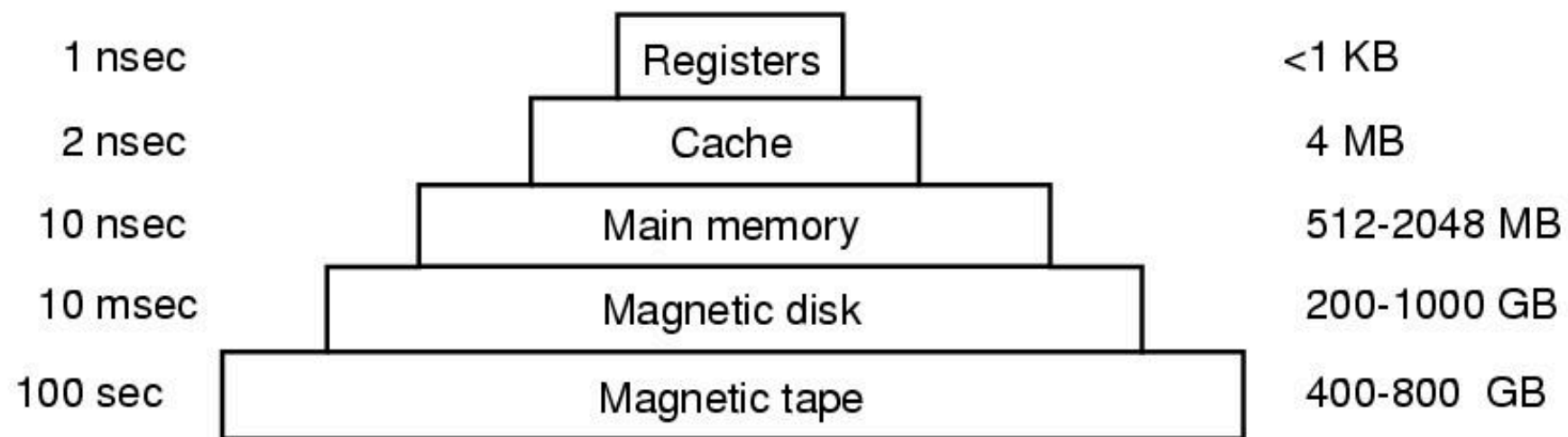
CHIPY WIELOWĄTKOWE I WIELORDZENIOWE



- (a) Czterordzeniowy układ scalony ze wspólną pamięcią cache L2.
- (b) Czterordzeniowy układ scalony z oddzielnymi pamięciami cache L2.

Typical access time

Typical capacity

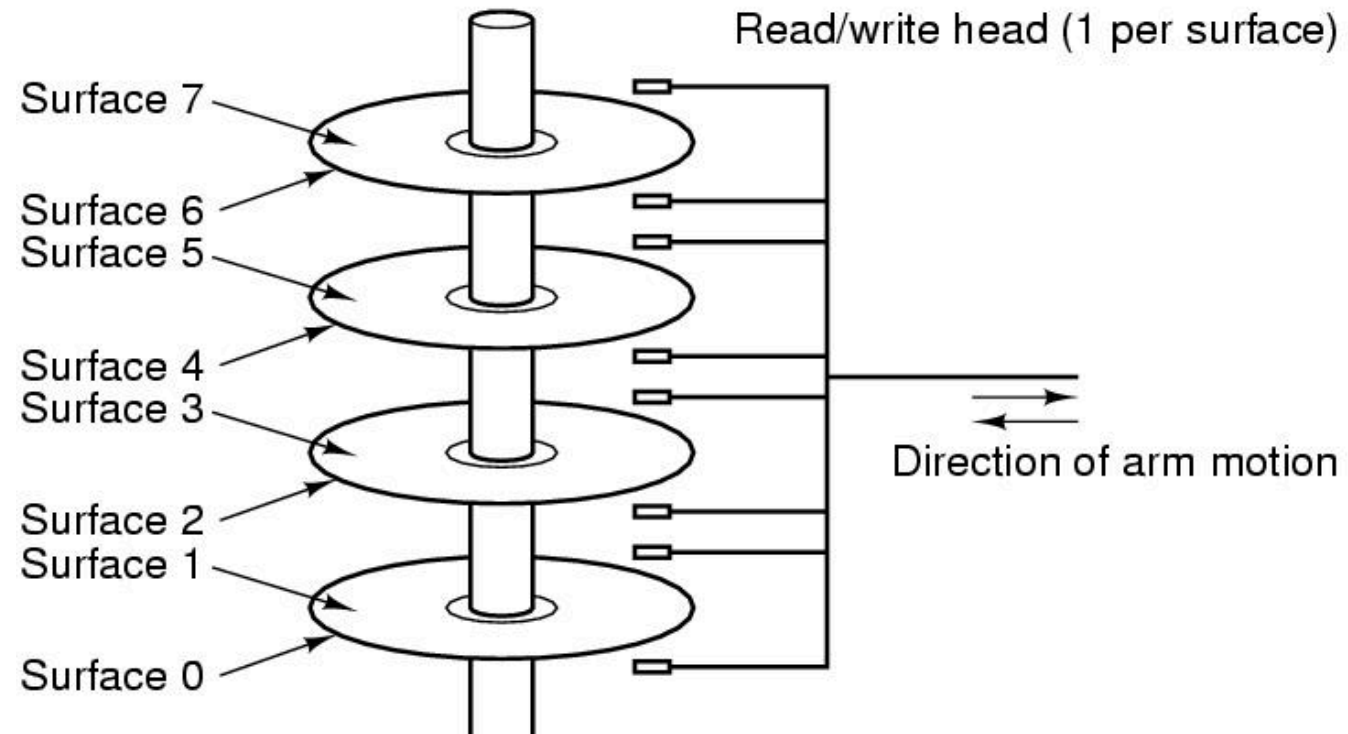


PAMIĘĆ

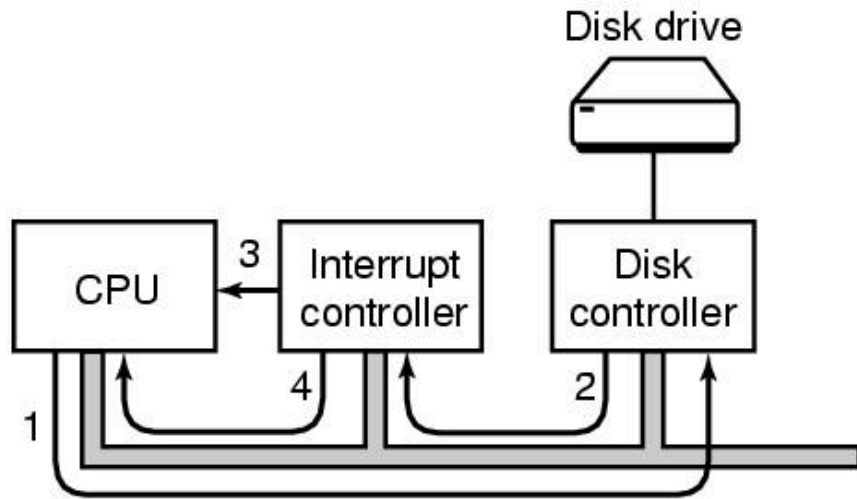
PAMIĘĆ

- Pytania dotyczące obsługi pamięci podręcznej na które OS musi znać odpowiedź:
 - Kiedy umieścić nowy element w pamięci podręcznej?
 - W której linii cache, umieścić nowy element?
 - Który element usunąć z pamięci podręcznej, gdy potrzebne jest nowe miejsce?
 - Gdzie umieścić nowo „wyeksmitowaną” daną w większej pamięci?

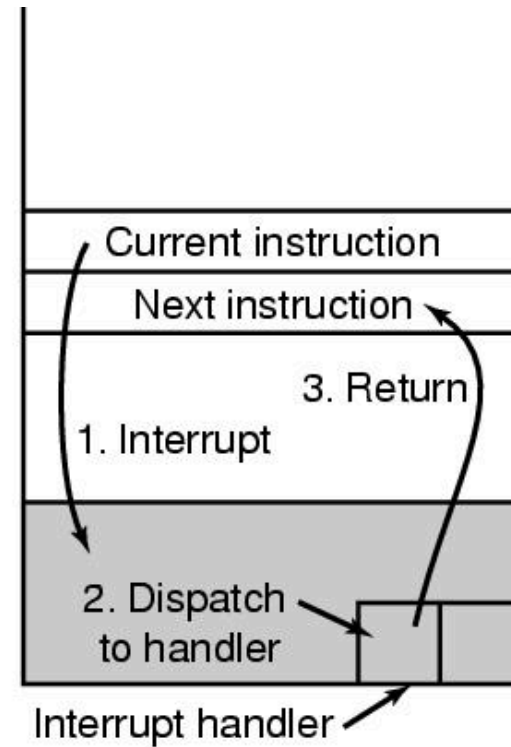
STRUKTURA DYSKU TWARDEGO



KROKI URUCHAMIANIA URZĄDZENIA WE / WY I PRZERWANIE.



(a)



(b)

CO NOWEGO W OPROGRAMOWANIU?

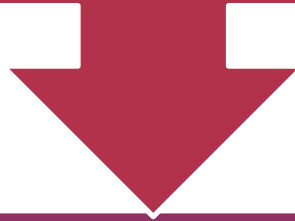
Wieloprocusowość symetryczna.

Era obliczeń równoległych.

Spopularyzowanie obliczeń czasu rzeczywistego, choć pierwsze tego typu systemy pojawiły się w 1964r.

Era komputerów personalnych – obniżenie kosztów sprzętu i oprogramowania.

Komputer analizuje dane dostarczane np. z czujnika i na ich podstawie reguluje działanie kontrolowanego obiektu, tak by zmieniły się dane odczytywane z czujnika. Często trzeba polegać na predykcji i wyznaczać przyszłe zachowania systemu.



Przykłady:

nadzorowanie
eksperymentów
naukowych,

obrazowanie
badań
medycznych,

sterowanie
procesami
przemysłowymi,

sterowniki:
wtrysku do
silników,
urządzeń AGD,

kontrola ruchu
powietrznego
lub urządzeń w
kosmosie.

SYSTEMY CZASU RZECZYWISTEGO

CECHY SO CZASU RZECZYWISTEGO

- System operacyjny musi mieć zdefiniowane stałe ograniczenie czasowe.
- Przetwarzanie danych musi się zakończyć przed upływem określonego czasu.
- Dane muszą być przechowywane w pamięci o bardzo krótkim dostępie.
- System operacyjny musi być w stanie udzielać szybkich odpowiedzi.

SYSTEMY ROZPROSZONE

Rozdzielanie obliczeń na wiele procesorów

- ściśle powiązane - współdzielenie pamięci i zegara (komunikacja przez pamięć dzieloną).
- luźno powiązane (systemy rozproszone) – każdy procesor ma własną pamięć lokalną (komunikacja przez szynę danych lub inne łącze).

Po co buduje się systemy rozproszone?

- Podział zasobów – rozproszone bazy danych, drukarki, specjalistyczne urządzenia.
- Przyspieszenie obliczeń – wykorzystanie mocy obliczeniowej wszystkich jednostek pośrednich.
- Niezawodność – unieruchomienie jednego komputera nie zaburza całego systemu.
- Łączność – poczta elektroniczna.

GRAPHICAL USER INTERFACE (GUI)



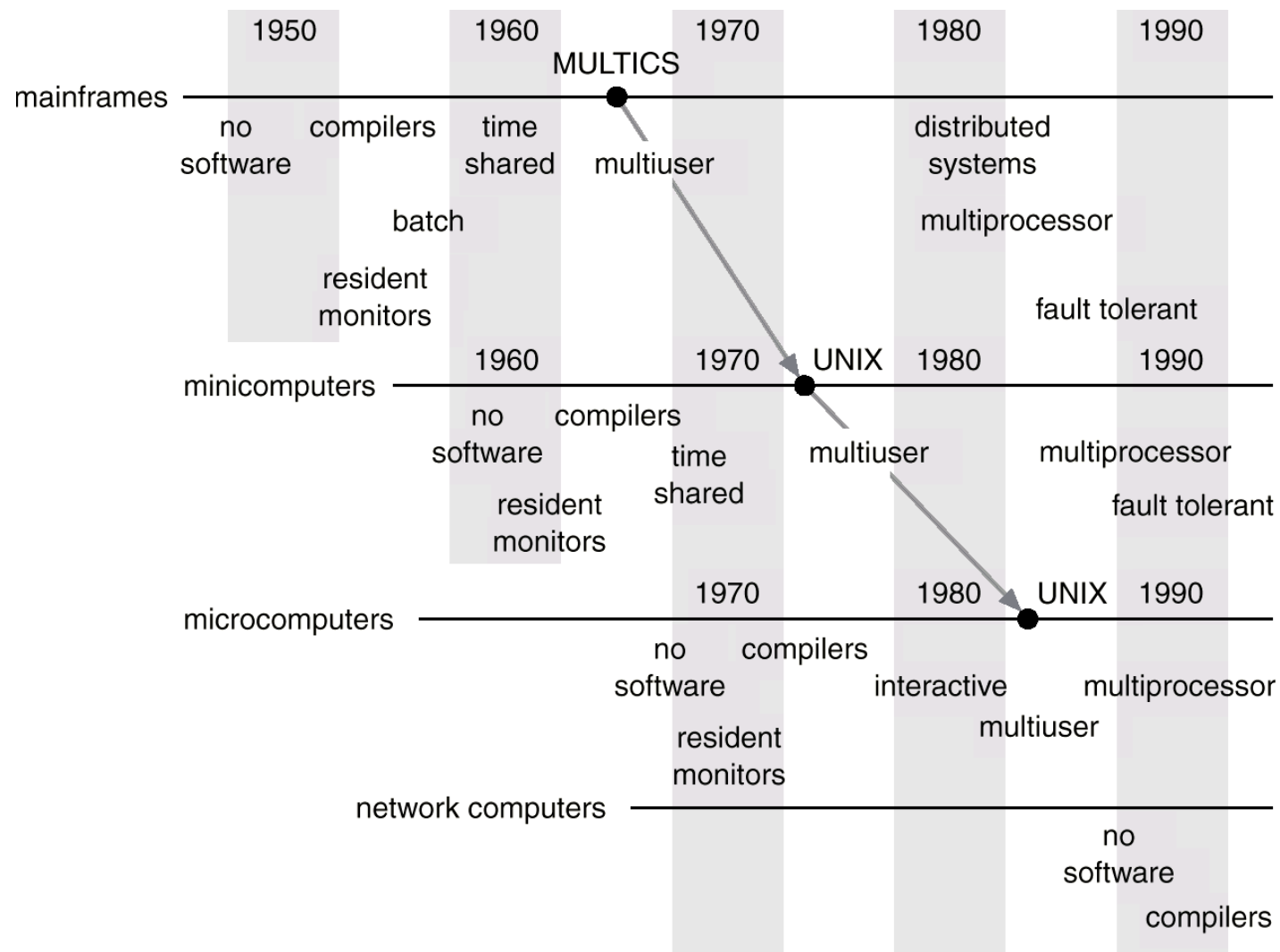
MICROSOFT®

Microsoft Windows
Version 1.01

Copyright (c) Microsoft Corporation, 1985. All Rights Reserved.
Microsoft is a registered trademark of Microsoft Corp.

COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK

- Interfejs wczesnych systemów operacyjnych to linia poleceń.
- Naukowcy z Xerox-PARC zbudowali pierwszy system operacyjny z graficznym interfejsem użytkownika.
- Steve Jobs odwiedził PARC, zobaczył GUI i użył go w Apple's Lisa (1983), później Macintosh (1984).
- Microsoft wprowadził system Windows w 1985 roku.
- <https://www.youtube.com/watch?v=J33pVRdxWbw>



MIGRACJA CECH I KONCEPCJI SYSTEMÓW OPERACYJNYCH

SYSTEMY OPERACYJNE RODZAJE

Systemy operacyjne
mainframe

Systemy operacyjne
serwera

Wieloprocessorowe
systemy operacyjne

Systemy operacyjne
komputerów
osobistych

Podręczne systemy
operacyjne

Wbudowane
systemy operacyjne

Systemy operacyjne
węzła sensorowego

Systemy operacyjne
w czasie
rzeczywistym

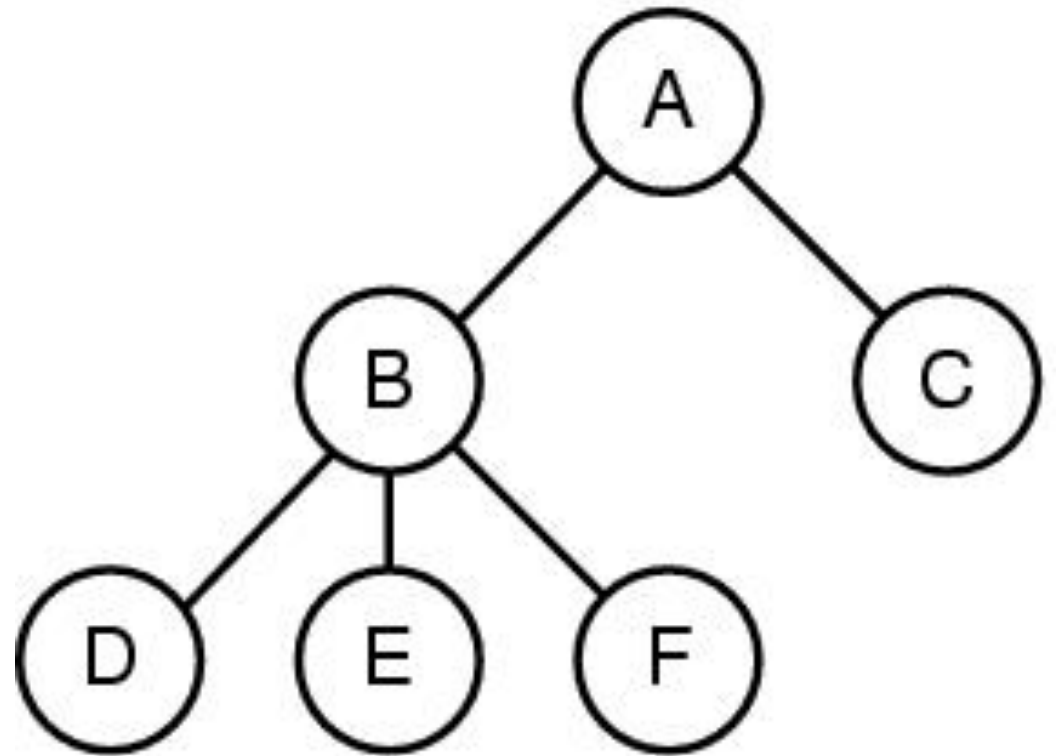
Systemy operacyjne
kart inteligentnych

POJĘCIA ZWIĄZANE Z SO

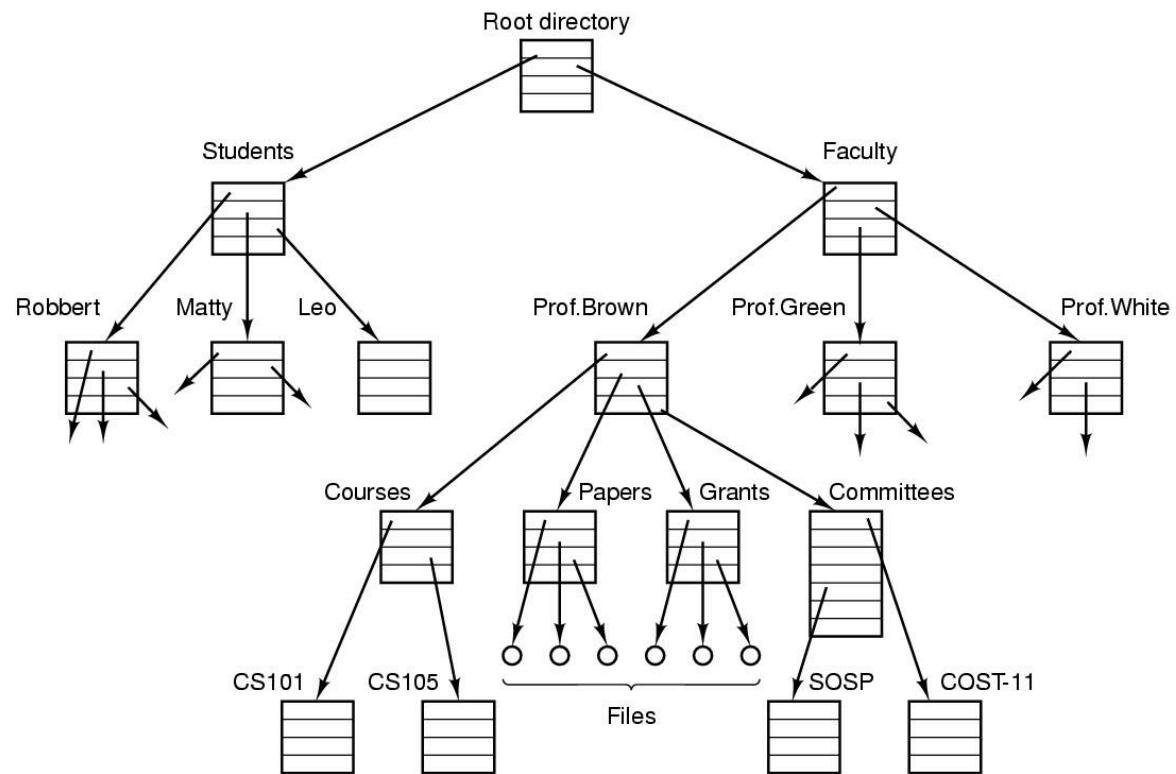
- Procesy
- Przestrzenie adresowe
- Pliki
- Wejście wyjście
- Ochrona
- Powłoka (shell)
- Ontogeneza odzwierciedla filogenezę
- Pamięci o dużej pojemności
- Zabezpieczania sprzętowe
- Dyski
- Pamięć wirtualna

PROCESY

- Drzewo procesów.
- Proces A stworzył dwoje dzieci procesy, B i C.
- Proces B stworzył trzy dzieci procesy, D, E i F.

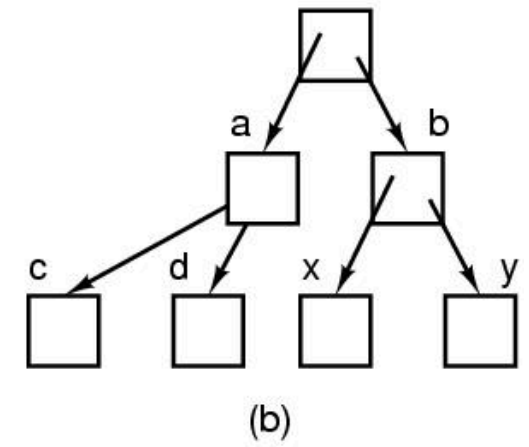
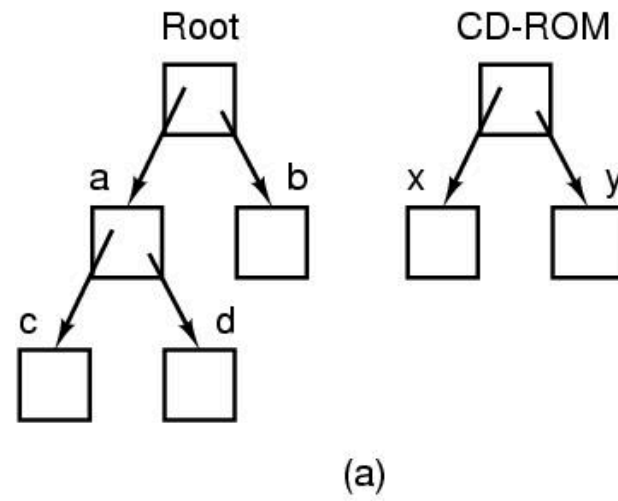


PLIKI – PRZYKŁADOWY SYSTEM PLIKÓW



PLIKI

- (a) Przed zamontowaniem pliki na płycie CD-ROM nie są dostępne.
- (b) Po zamontowaniu są częścią hierarchii plików.



POTOKI – PSEUDO PLIKI

COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK



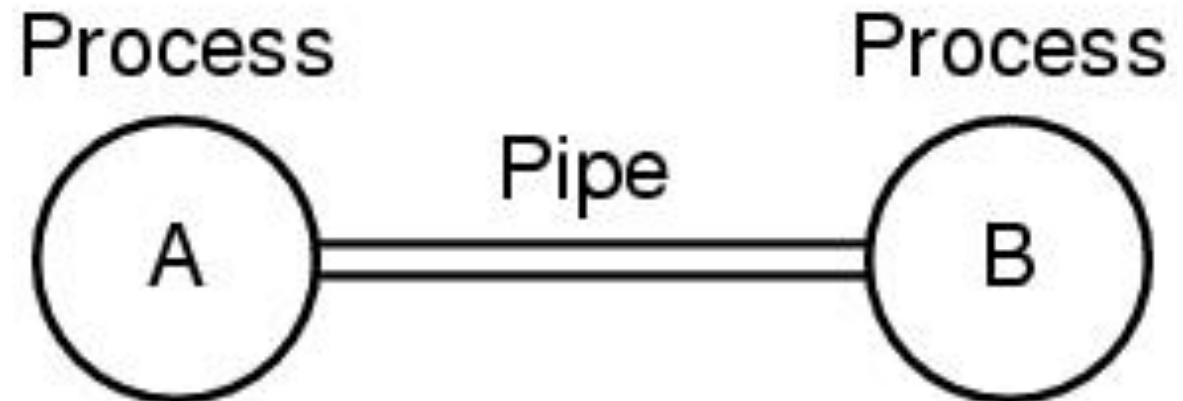
Dwa procesy chcą się ze sobą komunikować przez potok.

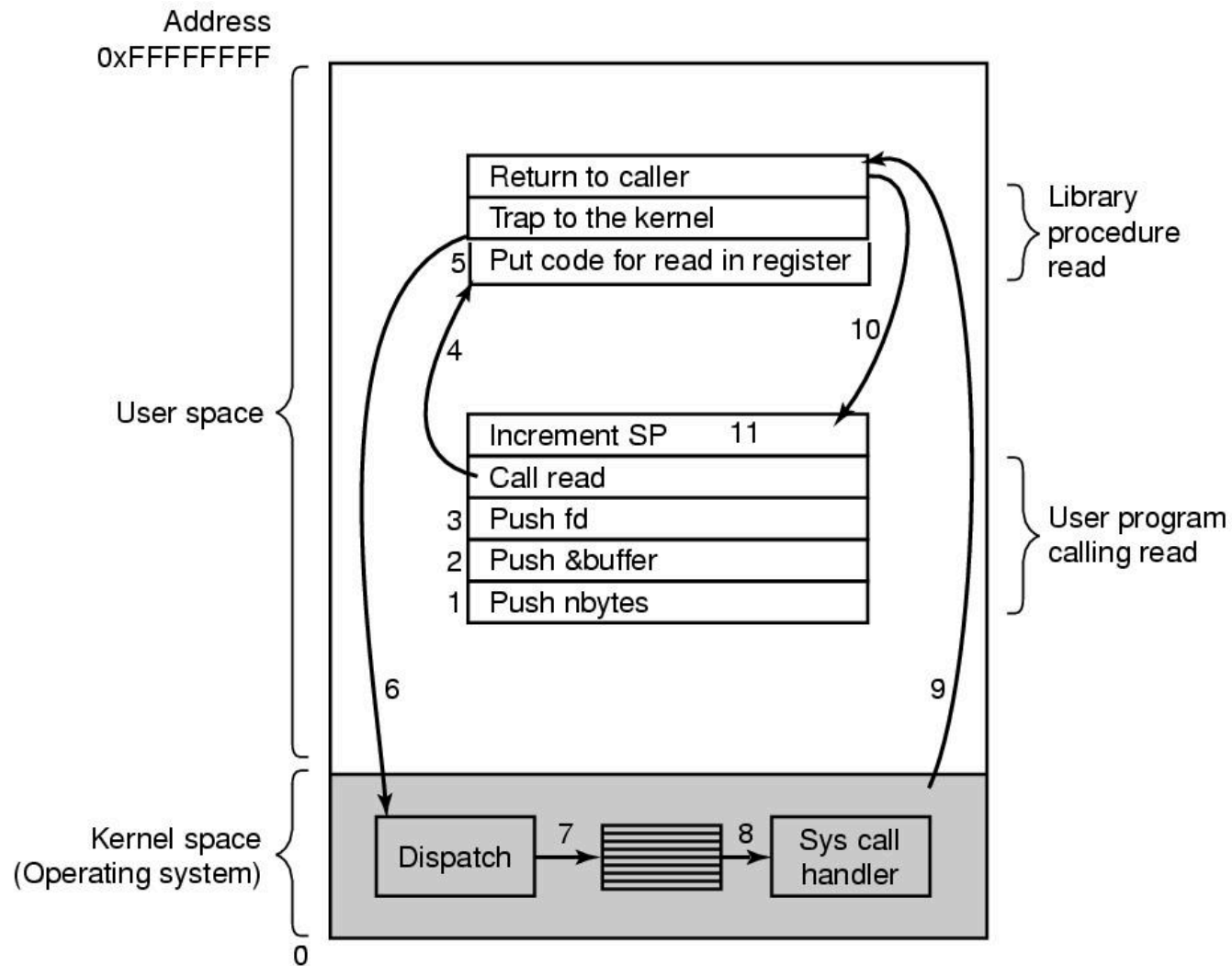


Muszą go ustanowić.



Jeżeli proces A chce przekazać dane do procesu B to zapisuje to w potoku.





COPYRIGHT: DR INŻ JOANNA KOŁODZIEJCZYK

WYWOŁANIA SYSTEMOWE

STRUKTURA SYSTEMÓW OPERACYJNYCH

- Systemy monolityczne - podstawowa struktura:
 - Główny program, który wywołuje żadaną procedurę serwisową.
 - Zestaw procedur serwisowych, które wykonują wywołania systemowe.
 - Zestaw procedur użytkowych, które pomagają w obsłudze procedury.

```

#define FALSE 0
#define TRUE 1
#define N      2          /* number of processes */

int turn;                /* whose turn is it? */
int interested[N];      /* all values initially 0 (FALSE) */

void enter_region(int process); /* process is 0 or 1 */
{
    int other;           /* number of the other process */

    other = 1 - process; /* the opposite of process */
    interested[process] = TRUE; /* show that you are interested */
    turn = process;      /* set flag */
    while (turn == process && interested[other] == TRUE) /* null statement */ ;
}

void leave_region(int process) /* process: who is leaving */
{
    interested[process] = FALSE; /* indicate departure from critical region */
}

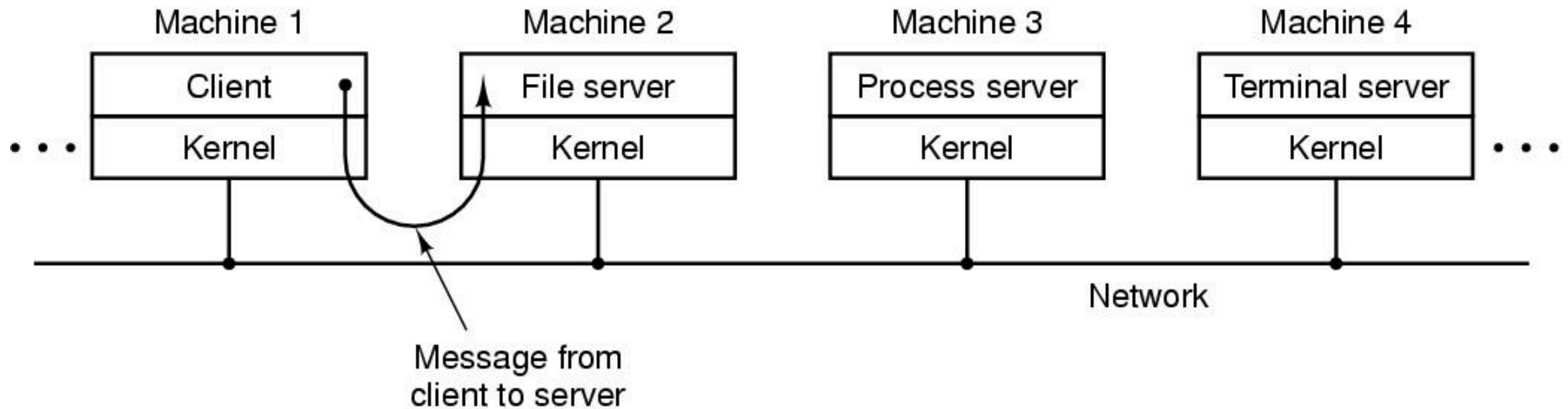
```

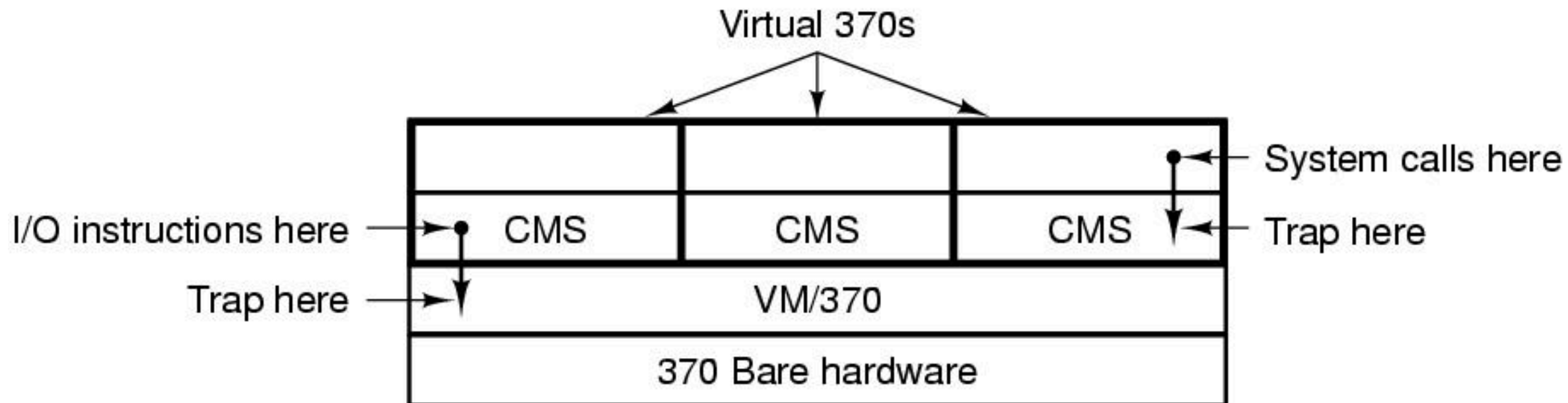
SYSTEM MONOLITYCZNY – PROSTA STRUKTURA

SYSTEMY WARSTWOWE

Warstwa	Funkcja
5	Operator
4	Programy użytkownika
3	Zarządzanie wejściem-wyjściem
2	Komunikacja pomiędzy operatorem i procesami
1	Zarządzenia pamięcią główną i bębnową
0	Przydział procesora i wieloprogramować

MODEL KLIENT-SERWER

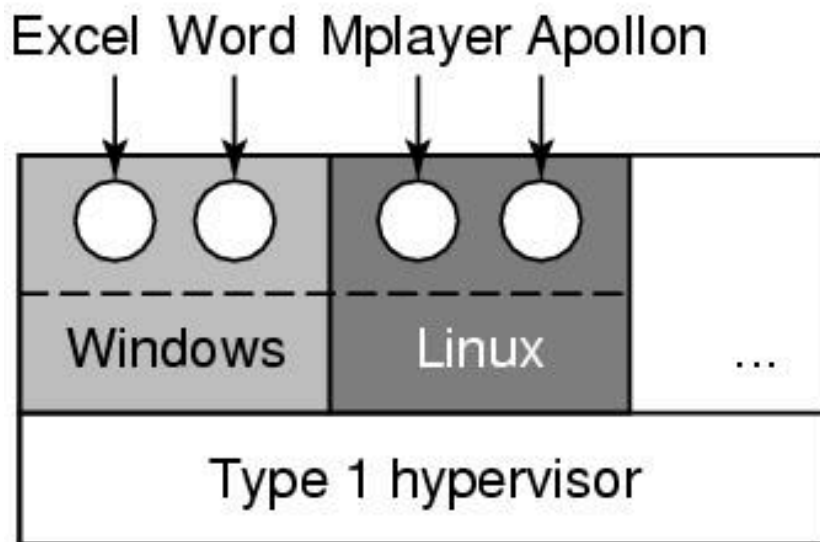




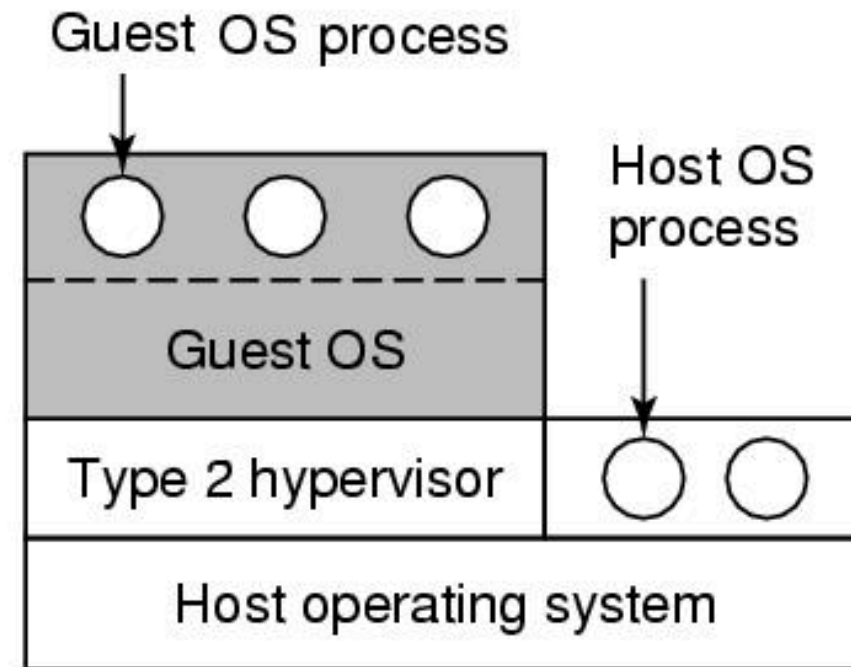
MASZYNY WIRTUALNE VM370 (1979)

MASZYNA WIRTUALNA

(A) A TYPE 1 HYPERVISOR. (B) A TYPE 2 HYPERVISOR.



(a)



(b)

MIKROJĄDRA – MINIX 3

