

Systemy operacyjne Wykład 01

Wersja 2024

dr inż. Marek Wilkus <http://home.agh.edu.pl/~mwilkus>
Wydział Inżynierii Metali i Informatyki Przemysłowej
AGH Kraków

Na podstawie programu opracowanego przez dr inż. Krzysztofa Wilka

1

Warunki zaliczenia

- Zaliczenie z ćwiczeń laboratoryjnych (wg warunków prowadzącego) – obecność!
- Pisemny egzamin z wykładów.
- Ocena końcowa: ćwiczenia (0.4) + Egzamin (0.6).
- Mamy 14 wykładów po 2 godziny akademickie.

2

Tematyka zajęć

- Ewolucja systemów operacyjnych.
- Rodzaje, właściwości i zadania systemów.
- Jądro systemu – budowa, funkcje.
- Procesy, wątki, współpraca między procesami, współistnienie.
- Zarządzanie (organizacja, adresowanie) pamięcią.
- System plików.
- Urządzenia wejścia-wyjścia.
- Systemy rozproszone.
- Systemy czasu rzeczywistego.
- Przykłady systemów i ich budowy.

3

Laboratoria

- System UNIX:
 - Działanie w systemie,
 - Poruszanie się w systemie plików,
 - Wykorzystanie narzędzi,
- Powłoka systemu:
 - Pisanie skryptów powłoki,
 - Automatyzacja zadań,
- Funkcje systemowe (opcjonalnie)
- Język PERL/Python/... (opcjonalnie)

4

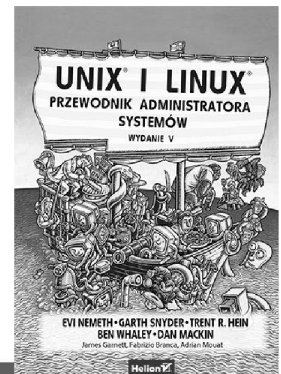
Literatura

- Silberschatz A. Galvin P.B. - Podstawy systemów Operacyjnych
- Nemeth E., Snyder G. Hein T. Whaley B. - Unix i Linux: Przewodnik administratora systemów, Helion 2011
- Lister A. M. Eager R. D. - "Wprowadzenie do systemów operacyjnych", WNT 1994
- Stevens R.W. - Programowanie w środowisku systemu UNIX, WNT 2002

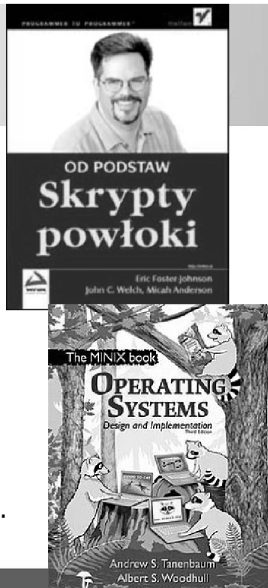
5

Literatura – c.d.

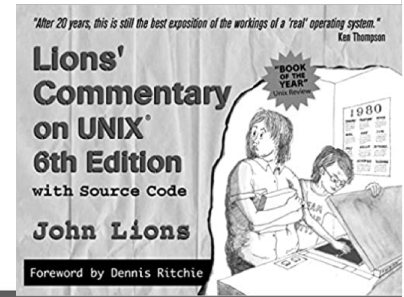
- Bach M.J. - Budowa systemu operacyjnego UNIX, WNT 1995
- Petersen R. - Arkana Linux.



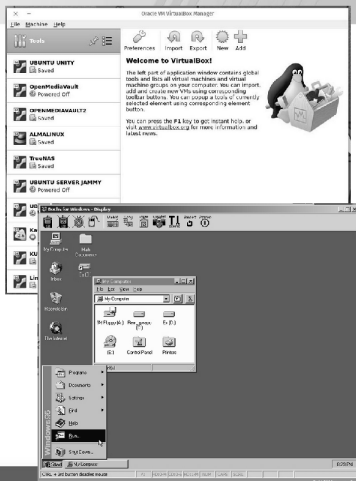
- Johnson E.F., Welch J.C., Anderson M. "Skrypty powłoki od podstaw", wyd. Helion, 2006
 - Praktyczny materiał do ćwiczeń z wykorzystania powłoki systemu, programowania Bash, Perl, sed, awk.
- Tanenbaum A.S., Woodhull A. - "Operating Systems: Design and implementation", 2006
 - Pozycja typu "Zrób to sam: uniksopodobny system operacyjny".



- "Podręcznik Lionsa", "Kod Lionsa" (<https://warsus.github.io/lions-/>)
 - Kompletny opis systemu Unix V6 wraz z porządnie komentowanym kodem źródłowym.
 - Uwaga: antyczny C.

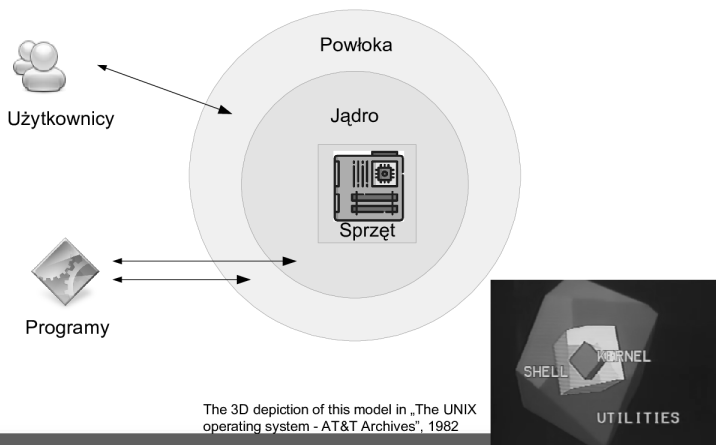


- VirtualBox – Program typu „Komputer w komputerze” – do testowania systemów operacyjnych o większych wymaganiach sprzętowych.
 - Dobry start jeżeli chcemy przetestować lub nauczyć się konfigurowania Linuksa.
 - Za mocne dla testowania starszych systemów.
- BOCHS – Emulatory PC do testowania starszych systemów.
 - PCem – ten emulator skupia się na wierności odtwarzanego sprzętu, głównie starszego.



- Definicja (wg A. S. i P. B. G.):

System operacyjny jest programem, który działa jako pośrednik pomiędzy użytkownikiem komputera a sprzętem komputerowym. Zadaniem systemu operacyjnego jest tworzenie środowiska w którym użytkownik może wykonywać programy.



- Jądro - komunikuje się z komputerem przez sterowniki urządzeń i wykonuje kolejowanie zadań, obsługę pamięci.
- Powłoka - stanowi interpreter poleceń systemu (komunikacja z użytkownikiem).
- Programy - polecenia systemowe nie zawarte w jądrze, programy narzędziowe, programy użytkowe.

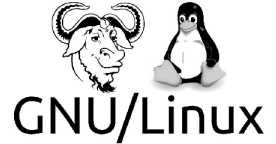
System operacyjny

- Głównym celem systemu operacyjnego jest to, aby był system komputerowy był **wygodny w użyciu**.
- Drugim celem jest **wydajna eksploatacja** sprzętu komputerowego.

13

Kilka pojęć na początek

- UNIX - Wielozadaniowy, wielodostępny system operacyjny, pierwotnie opracowany jako jądro + zestaw programów, rozwijanych głównie na uniwersytetach. Dostępny komercyjnie w różnych dystrybucjach sprzedawany przez różne firmy (Solaris, IBM AIX, Tru64 UNIX, HP-UX).
- GNU - wolna implementacja systemu UNIX-owego oparta w zamierzeniu na jądrze Hurd.
- Linux - Jądro systemu na wolnej licencji. Jeżeli działa na nim ekosystem GNU, to pełna nazwa (której raczej nikt nie używa) to GNU/Linux.
- BSD (Berkeley Software Distribution) - Kolejna wolna implementacja systemu uniksowego, oparta na własnym jądrze systemu.



14

Dystrybucje

- Dystrybucje to zestawy jądro + system bazy + programy. Takimi dystrybucjami GNU/Linuksa będzie np. Debian, Arch, CentOS, Ubuntu, a systemu BSD FreeBSD, NetBSD, PC-BSD.
- Czy można zrealizować np. GNU/BSD, czyli ekosystem GNU na jądrze BSD? Projekt Debian przeprowadził taki eksperyment, lecz korzyści wydajnościowe nie były zbyt duże.
 - Projekt Nexenta: GNU na jądrze OpenSolarisa.
 - Cygwin - ekosystem GNU działa na Windowsach.

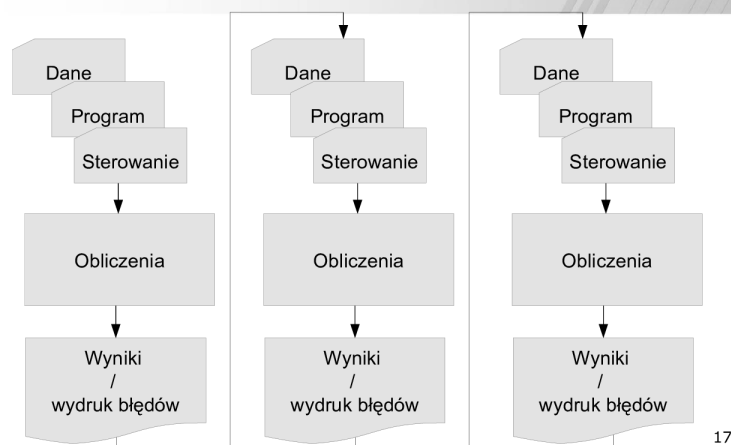


15

Ewolucja systemów operacyjnych

16

Systemy wsadowe



17

Systemy wsadowe

- Zadania wykonywane są kolejno.
- Każde zadanie składa się z wczytywania programu i danych, obliczeń oraz zapisu (lub wydruku) wyników.
- Następne zadanie wykonywane jest po zakończeniu poprzedniego.
- Kolejność zadań ustalana przez operatora.
- Zadania o podobnych wymaganiach grupowane są w tzw. wsad (batch).

18

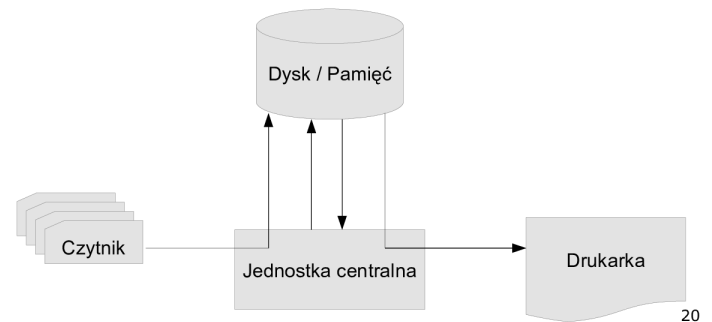
Systemy wsadowe

- Zalety:
 - Bardzo prosty system, tylko automatyczne przekazywanie sterowania od jednego zadania do drugiego.
- Wady:
 - Brak nadzoru użytkownika podczas wykonywania zadania (niemożność podjęcia nawet prostych decyzji, interakcji).
 - Tylko jedno zadanie w danym czasie.
 - Duża bezczynność jednostki centralnej podczas wczytywania i wydruku danych (obliczenia: tysiące operacji / s, wczytywanie: kilka/kilkanaście kart / s).

19

SPOOLing

- Simultaneous Peripheral Operation On-Line



20

Spooling

- W tym samym czasie wykonywane są obliczenia jednego zadania, i operacje I/O innego.
- Kosztem zajęcia niewielkiej części pamięci stało się możliwe efektywniejsze wykorzystanie zasobów CPU i urządzeń peryferyjnych.

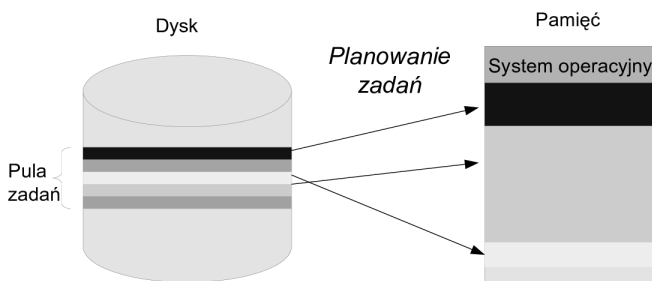
21

Spooling

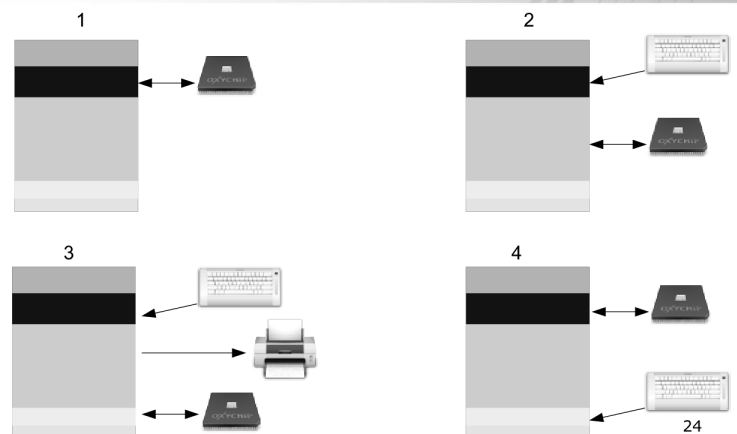
- Zalety:
 - Lepsze wykorzystanie jednostki centralnej.
 - Minimalizacja "przebiegów" urządzeń wejścia/wyjścia.
 - Możliwość pracy programu i urządzeń I/O w tym samym czasie.
- Wady:
 - Konieczność wykorzystania pamięci.

22

Wieloprogramowy system wsadowy



23



24

Wieloprogramowy system wsadowy

- W tym samym czasie system operacyjny przechowuje w pamięci kilka zadań.
- Gdy aktualnie wykonywane zadanie oczekuje na usługę lub zakończenie operacji (np. I/O), wykonywane jest następne zadanie.
- Gdy zakończyło się oczekiwanie, a następne zadania są zajęte, system powraca do wykonywania poprzedniego zadania.

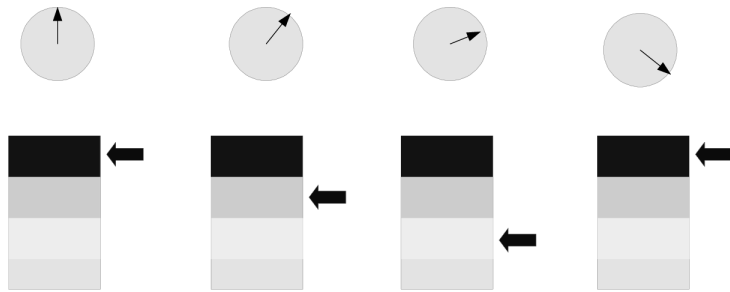
25

Wieloprogramowy system wsadowy

- Zalety:
 - Lepsze wykorzystanie procesora i urządzeń I/O.
 - Szybsze wykonywanie puli programów.
- Wady:
 - Wymaga skomplikowanego systemu operacyjnego (planowanie zadań, przydział procesora).
 - Zadanie obliczeniowe, bez operacji I/O, może na długo zablokować resztę zadań.

26

System z podziałem czasu



27

System z podziałem czasu

- Procesor wykonuje wiele różnych zadań na przemian.
- Przełączanie następuje tak często, że użytkownicy mogą współdziałać z każdym programem podczas jego wykonywania.
- Każdy z użytkowników odnosi wrażenie, że dysponuje własnym komputerem, choć jest on w rzeczywistości współdzielony.

28

Historia popularnych systemów

- CP/M (G. Kildall, 197x-8x) - tekstowy, jednozadaniowy, jeden użytkownik...
 - ...można udawać, że wielu użytkowników partycjonując dyskietkę, ale i tak nie ma żadnych zabezpieczeń...
 - Z niego wyewoluował MS-DOS.
- MS-DOS
 - 08.1980: Tim Patterson, Seattle Computer Products - tworzy prototyp,
 - 12.1980: QDOS, Microsoft wykupuje prawa bez wyłączności.
 - 1981: Zaadaptowany do IBM PC jako MS-DOS 1.0
 - 07.1981: MS wykupuje wszystkie prawa od SCP za \$50 000. 86-DOS 1.0

29

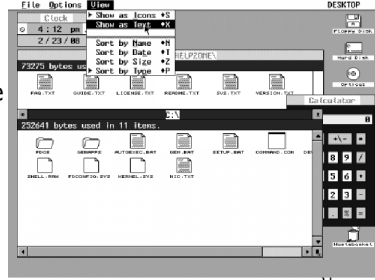
DOS (c.d.)

- Gdy Microsoft jeszcze nie wykupił wszystkich praw, sprawą zajął się równolegle IBM...
 - PC-DOS 1.0 i 1.05 - działa z dyskietkami 160kB. Posiada spójny interpreter poleceń COMMAND.COM z pełną obsługą operacji plikowych (nie jest wymagany zewnętrzny program PIP jak w CP/M).
- 1983: PC-DOS 2.0, wspiera katalogi i strumienie I/O, dostęp do plików przez identyfikatory (handle) i sterowniki urządzeń. Dyskietki 320 i 360kB.
- 1984: DOS 3.0 i 3.3 - 16-bitowy FAT, wsparcie PC-AT, dyski do 20MB (więcej przez zewnętrzne sterowniki), dodatkowe narzędzia. Dyskietki high-density (1.2MB, 1.44MB)
- DOS 6.22 - ostatni DOS oferowany jako osobny produkt.

30

DOS - lata 90-te - zakończenie

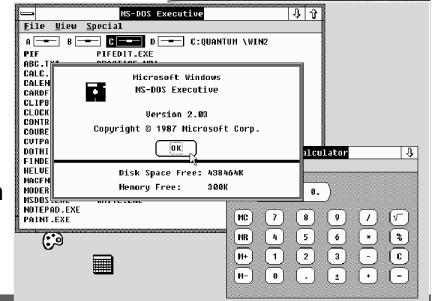
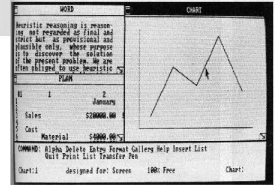
- DOS 7.0 - część Windowsów 9x.
- Caldera DOS, DR-DOS, MUDOS, DOS Plus - alternatywy.
- FreeDOS - otwartoźródłowy system zgodny z MS/PC DOS.



- Otwierają źródła środowiska graficznego GEM, umożliwiającego uruchamianie kilku programów jednocześnie (program w tle jest wstrzymywany).

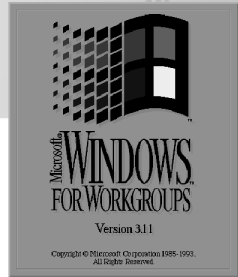
MS Windows "Visi-On wygląda ciekawie, zróbmy coś podobnego"

- 1984 - Windows preview - jeszcze w dużej mierze semigrafika.
- 06.1985 - Windows 1.0, 1.01,
- 1987 - Windows 2.0 i Windows/286 - okienka już można na siebie nakładać, wsparcie dla wielozadaniowości. Separacja programów, Standard Windowsowych plików EXE.



MS Windows

- 1990 - Windows 3.0 - wymaga ≥ 640 kB RAMu. Lepsze GUI i praca z wieloma programami. Interoperacyjność (OLE, kopiuj-wklej w całym systemie).

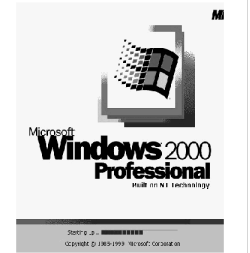


- 1993 - Windows for Workgroups 3.11 - praca w sieci na poziomie programów systemu. Obsługa multimediów i dynamicznie ładowanych sterowników.



Windows - c.d.

- 1994 - Windows NT - podejście od nowa bez balastu MS-DOS.
- 1995 - Windows 95 - ściślejsza współpraca między programami, DOS jest prawie zawsze w tle.
- 1996 - Windows 95 OSR2 - „Oddajcie mi przeglądarkę!”.
- 1997 - Windows NT 4.0 - profesjonalny system do zastosowań na stacjach roboczych i w sieci.
- 1998 - Windows 98.
- 02.2000 - Windows 2000 - czyli NT 5.0.
- 09.2000 - Windows ME (Millennium) - Ostatni Windows z DOSem w tle.



Windows (c.d.)

- 10.2001 - Windows XP (NT 5.1) - wym. 300MHz, 128MB RAM, 1.5GB HDD, wraz z kolejnymi aktualizacjami więcej.
- 2003 - Windows Server 2003.
- 2005 - Windows Vista,
- 2007 - Windows 7,
- 2012 - Windows 8,
- 2014 - Windows Server 2012R2,
- 2015 - Windows 10.



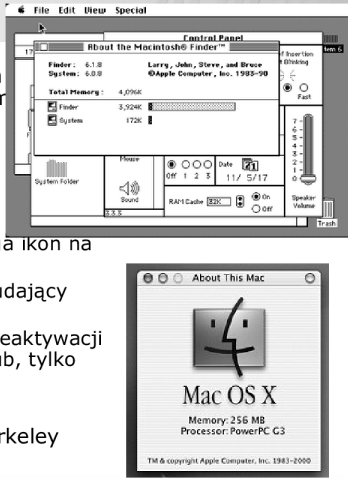
ReactOS

- W Linuksie dzięki projektowi **WINE** możliwe jest uruchamianie programów windowsowych. WINE to napisane od nowa API systemu Windows dla innych platform.
- A gdyby tak dodać do tego jądro, sterowniki, zarządzanie pamięcią i zrobić otwartoźródłowy Windows?
- Projekt ReactOS (1996 - FreeWin95) - aktualnie w stanie wczesnej wersji alpha.



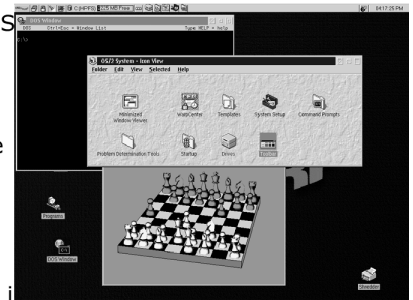
Inne systemy: Mac OS

- Powstał jako system dla komputera Macintosh 128K pod nazwą „System Software”, później Mac OS.
- Składniki:
 - Jądro - monolityczne, później z ładowalnymi modułami, jako interfejs do **toolbox ROMu**.
 - Finder - menedżer plików i obsługa ikon na pulpicie,
 - Disk Utility - sterownik dyskowy udający program narzędziowy.
 - Programy: Multipleksowane (po deaktywacji ich okna nie dostają procesora) lub, tylko niektóre, działające w tle.
- Rozwijany do wersji 9.2.2
- Później: Mac OS X, na postawie Berkeley Unixa.



Inne systemy: OS/2 (lata 90-te)

- Rozwijany przez IBM jako IBM DOS + Presentation Manager (graficzna nakładka).
- Później IBM ???dostał??? od MS pełny kod Windowsów 3.x i do wersji Warp 4 wbudowano API Windowsów.
- Znany z bardzo wydajnej, niskopoziomowej (na poziomie jądra!) implementacji Javy.
- W 1996 „podłączono” powłokę i... przeniesiono to na PowerPC. Działało. Nie było stabilne.



Inne systemy: BeOS

- Stworzony przez Be Inc. ok. 1995 dla komputerów multimedialnych BeBox.
- „Poligon doświadczalny” dla systemów wykorzystujących komponenty multimedialne.
- Pierwszy planista CPU wykorzystujący heurystykę.
- Przydził pamięci dla programów oparty o używane urządzenia.
- Ostatnia wersja R5 ok. 2001. Później R5 Dan0 (wyciek kodu wersji 6) i ZetaOS.



- Otwarta implementacja: Haiku OS (aktualnie w wersji beta)

UNIX



Denis Ritchie

Ken Thompson

Richard Stallman

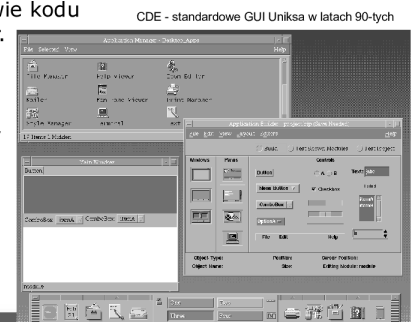
Linus Torvalds

UNIX

- 1966 - Rozpoczęcie prac nad multipleksowanym systemem MULTICS (Bell Labs) - K. Thompson
- 1969 - Pierwszy Unix w AT&T/Bell Labs, D. Ritchie, K. Thompson
- 1971 - Port Uniksa na PDP-11 - komputer, na który mógł pozwolić sobie uniwersytet,
- 1973 - Unix 4th edition - napisany w C.
- 1975 - Unix 6th Edition - rozprowadzany bezpłatnie na uniwersytetach i tam rozwijany. Początek BSD (Berkeley Unix)

UNIX (c.d.)

- 1977 - Pierwsze taśmy BSD,
- 1980 - SCO Unix, Microsoft XENIX, Coherent UNIX Preview.
- 1981 - DEMOS - Wysoce zoptymalizowany Unix o jeszcze niższych wymaganiach opracowany w ZSRR na podstawie kodu BSD przepisane na assembler.
- 1982 - Silicon Graphics IRIX,
- 1983 - **GNU (R. Stallman)**
- 1984 - Hewlett-Packard HP-UX,
- 1990 - IBM AIX,
- 1991 - Sun Microsystems: **Solaris 2**
- **1991 - Linux 0.01 (L. Torvalds)**

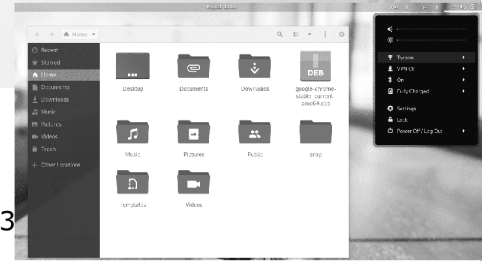


- 1993 - FreeBSD 1.0
- 1993 - **Debian GNU/Linux 0.x**,
- 1994 - Red Hat Linux, Caldera Linux, początek różnych dystrybucji,
- 13.03.1994 - **Linux 1.0**
- 1996 - Linux 2.0
- 1999 - Linux 2.2
- 2001 - Linux 2.4,
- 2002 - Środowisko graficzne KDE 3,
- 2003 - FreeBSD 5.0,
- 2008 - OpenSolaris,
- 2010 - Koniec rozwoju OpenSolaris.
- 2011 - Oracle Solaris 11

*Przykład Linuksa z KDE3,
(ekran na komputerze u Prowadzącego)*

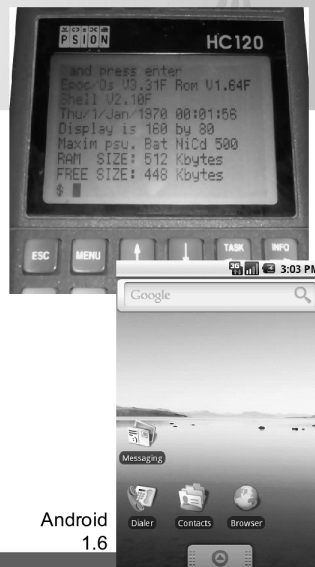
43

- 2015 - Oracle Solaris 11.3
- 2015 - FreeBSD 10.3
- 05.2016 - Linux 4.6
- 08.2016 - **Android 7.0**
- 08.2017 - Android 8.0
- 2015-17 - Uniksowy system uruchomieniowy INIT zostaje w Linuksach stopniowo zastąpiony przez systemd. Pojawiają się binarne logi, zrównoleglone uruchamianie systemu i złożone pliki konfiguracyjne
- 2020 - Linux 5.x, 2022 - Linux 6.x



44

- 198x - PAL (DOS 3.x + ekstrakody)
- 199x - EPOC → Symbian
- Android:
 - 1.x (2008) - pierwotnie projektowany jako system dla aparatów cyfrowych.
 - 2.x (2009) - Interfejs jako dokument dla przeglądarki.
 - 3.x (2011) - Wydajniejsze GUI, wsparcie dla wielu rdzeni i systemów plików.
 - 4.x (2011)
 - 5.x (2014) - Dynamicznie ładowane sterowniki z przestrzeni użytkownika (widoczne jako np. Działające karty dźwiękowe na USB)



Android
1.6

- BSD + mechanizm kitów API (BeOS) + fragmentacja usług → iOS (znaczną energooszczędność i większą wydajność).
- 2022 - Android używa coraz więcej kodu w Rust. Większe bezpieczeństwo kosztem wydajności i użycia pamięci.
 - Docelowo jądro Linux ma używać tej samej techniki.
- 2022 - Wciąż są rozwijane systemy oparte na pojedynczych zadaniach lub przerwaniach (qronOS, OmnOS, SyMobi) - w zamkniętych zastosowaniach „feature phone” i przemysłowych.

46

Dziękuję za uwagę

47