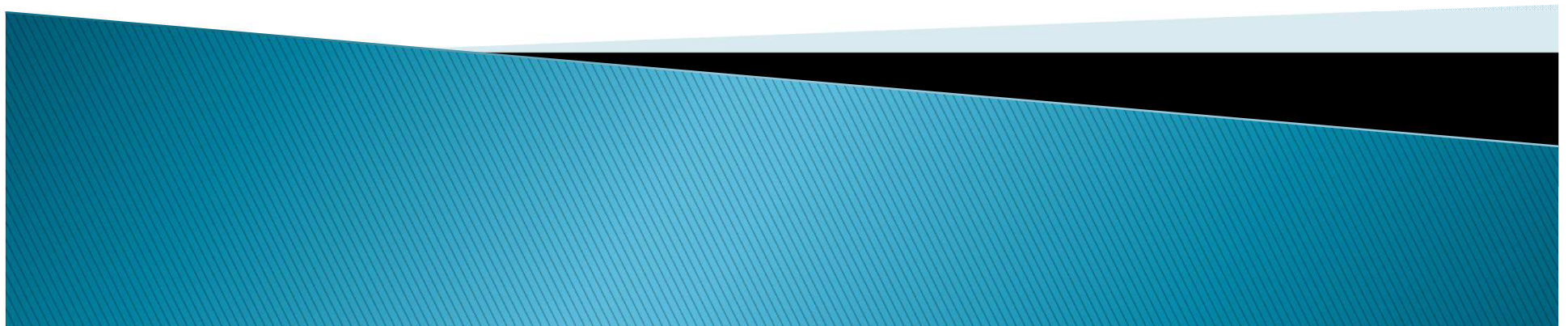


Оперативни Системи



Цел на предметот

- ▶ Запознавање со основните концепти на оперативните системи
 - UNIX/Linux
 - Windows

Цел на предметот

- ▶ Практична примена
 - Скрипти
 - Системски повици
 - Перформанси

Содржина на предметот

- ▶ ОС – општи поими, дефиниции и историја
- ▶ Концепт за системски ресурси
- ▶ Процеси и нитки
- ▶ Управување со меморија
- ▶ Датотечни системи
- ▶ Влез/ Излез
- ▶ Имплементации кај
 - UNIX/Linux и
 - Windows 2003/XP/Vista

Полагање на предметот

- ▶ Часови: 2+2+2
- ▶ Оценување
 - Лабораториски вежби (15%)
 - Писмен испит (45%)
 - Устен испит (40%)
- ▶ Оцени

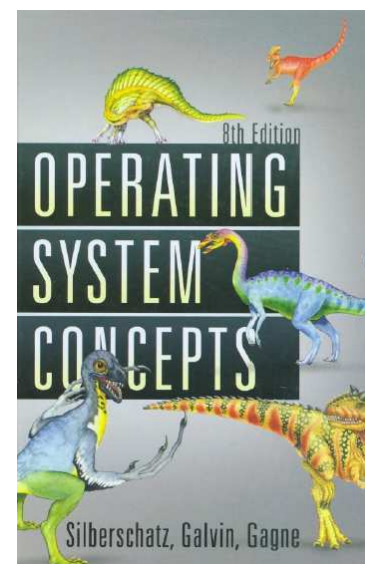
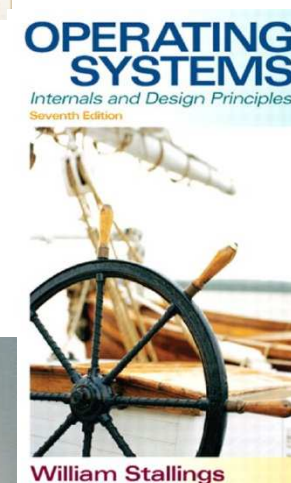
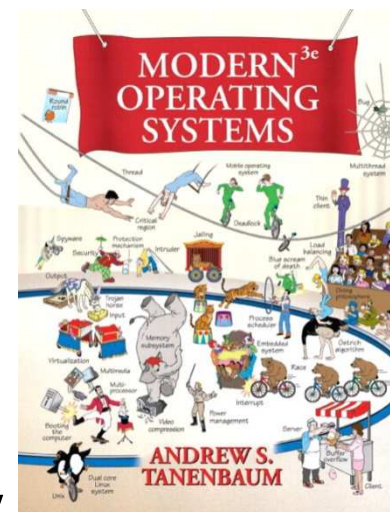
◦ 50 – 60	6 (шест)
◦ 61 – 70	7 (седум)
◦ 71 – 80	8 (осум)
◦ 81 – 90	9 (девет)
◦ 91 – 100	10 (десет)

Полагање на предметот

- ▶ Потпис се добива со максимум еден минус од лабораториски вежби
- ▶ Писмени и усно преку колоквиуми
 - Поените на I и II колоквиум, поединечно $\geq 40\%$
 - Збирот на поени да е $\geq 50\%$
- ▶ Писмено и усно на испит
 - $\geq 50\%$

Литература

- ▶ Andrew S. Tanenbaum , *MODERN OPERATING SYSTEMS*, 3rd–edition, Prentice Hall, 2008
- ▶ William Stallings, *OPERATING SYSTEMS, Internals and Design Principles*, 7th Edition, Prentice Hall
- ▶ Silberschatz, Galvin, Gane, *OPERATING SYSTEM CONCEPTS*, 8th Edition, Wiley, 2009.



Пресметувачките уреди се насекаде

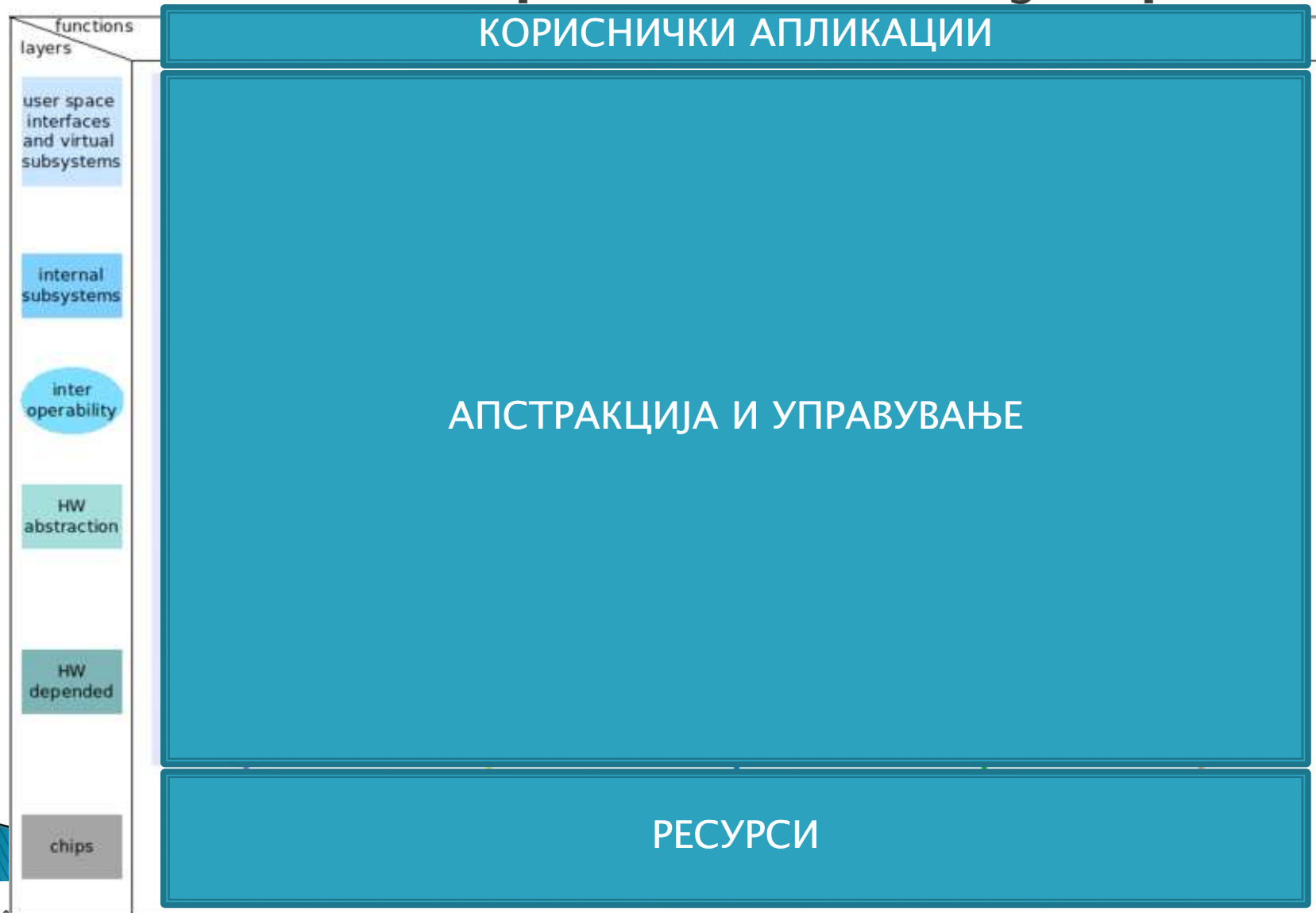


ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Ресурси

- ▶ ОС се управувачи со ресурси!
- ▶ Процесори
- ▶ Меморија
- ▶ В/И уреди
- ▶ Комуникациски уреди
- ▶ Податоци
- ▶ ...

Едноставен приказ на ОС јадро



Што прави еден ОС?

- ▶ Silberschatz и Gavin: “ОС е сличен на една влада”
 - Се наложува прашањето: дали владата прави нешто корисно како таква?
- ▶ Координатор и сообраќаец:
 - Управува со сите ресурси
 - Ги решава конфликтните барања за ресурсите
 - Штити од грешки и неправилно користење на компјутерот
- ▶ Олеснувач (“корисна” апстракција):
 - Обезбедува услуги и олеснувања потребни за сите
 - Стандардни библиотеки, Систем на прозорци
 - Го прави програмирањето на апликации полесно, побрзо и со помалку грешки

- ▶ Некои особини се засегнати од двете задачи:

- Податочниот систем е потребен на сите (Олеснувач) ...

- ... но податочниот систем мора да биде заштитен (Сообраќаец)

Што навистина претставува ОС?

▶ Најчесто:

- Управување со меморија
- Управување со I/O
- CPU распоредување
- Синхронизација / Примитиви за заемно исклучување
- Комуникација? (Дали Email програмата е дел од ОС?)
- Multitasking/multiprogramming?

▶ А што со?

- Податочниот систем?
- Поддршка за мултимедиа?
- Кориснички интерфејс?
- Интернет прегледник?



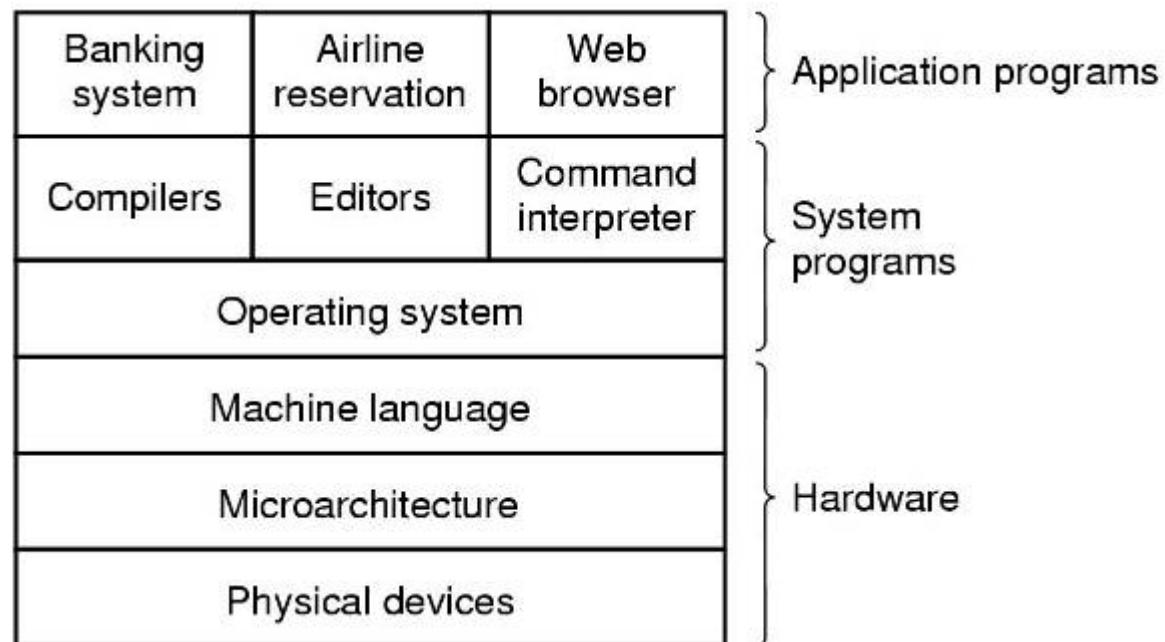
Дефиниции за ОС

- ▶ Не постои универзално прифатена дефиниција
- ▶ “*Се што дилерот ќе испорача, кога ќе нарачате оперативен систем*” е добра апроксимација
 - Но варира
- ▶ “*Единствената програма која се извршува постојано на еден компјутер*” е јадрото
 - Сето останато е системска програма (која се испорачува со оперативниот систем) или е апликациска програма

Главни особини на ОС

- ▶ **Да се скријат деталите за хардверот – апстракција**
 - Софтвер кој ги крие деталите на пониско ниво. ОС го трансформира физичкиот свет на уреди, инструкции, меморија и време во виртуелен свет
- ▶ **Алоцирање ресурси за процеси (управување со ресурси)**
 - ОС контролира како процесите (активните агенти) достапуваат до ресурсите (пасивните ентитети).
- ▶ **Овозможува пријатен и ефикасен кориснички посредник (интерфејс)**
 - Интерпретер на команди, датотечен систем, помош, апликациска интеграција...

Поставеност на ОС



- ▶ Компјутерскиот систем се состои од
 - Хардвер
 - Системски програми
 - Апликации

Што е оперативен систем (ОС)

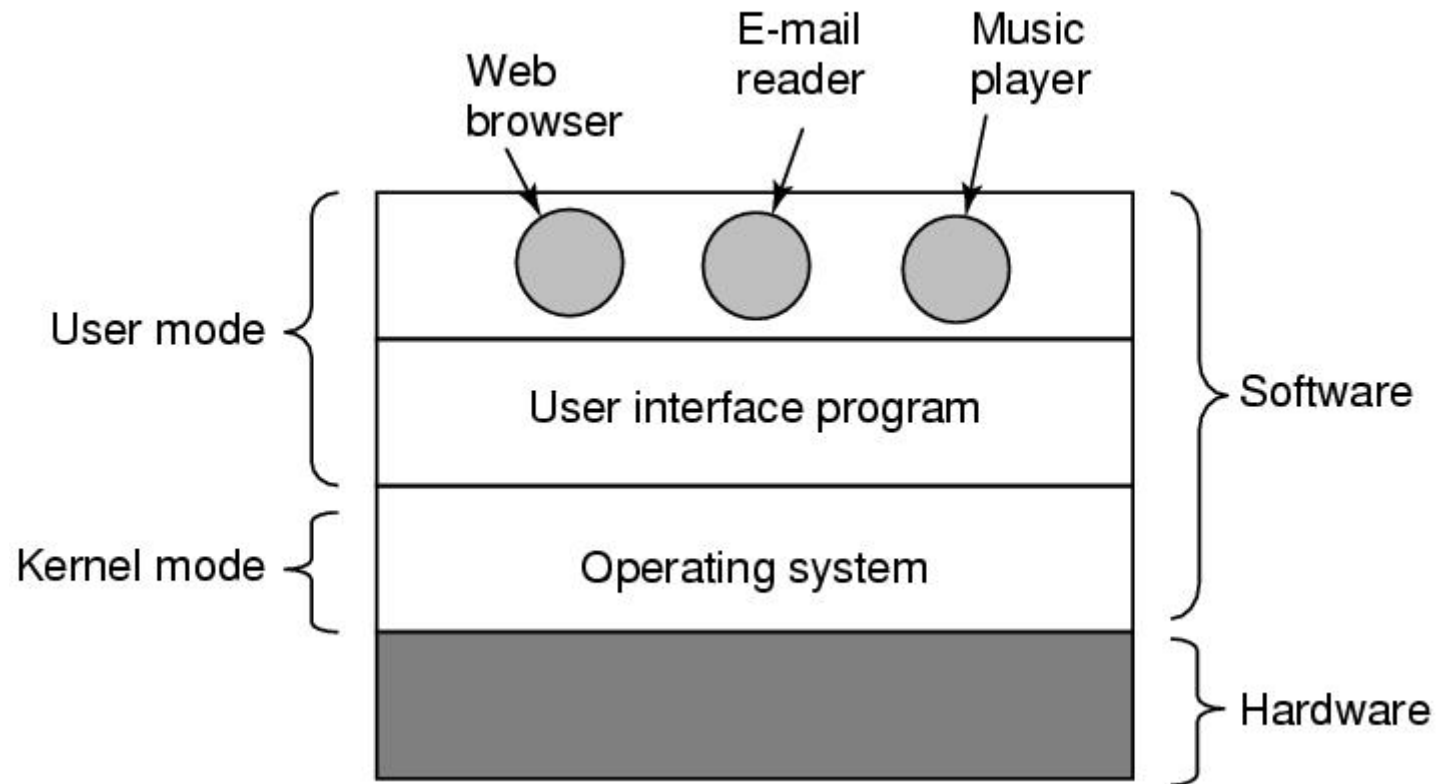


Figure 1-1. Where the operating system fits in.

The Operating System as an Extended Machine

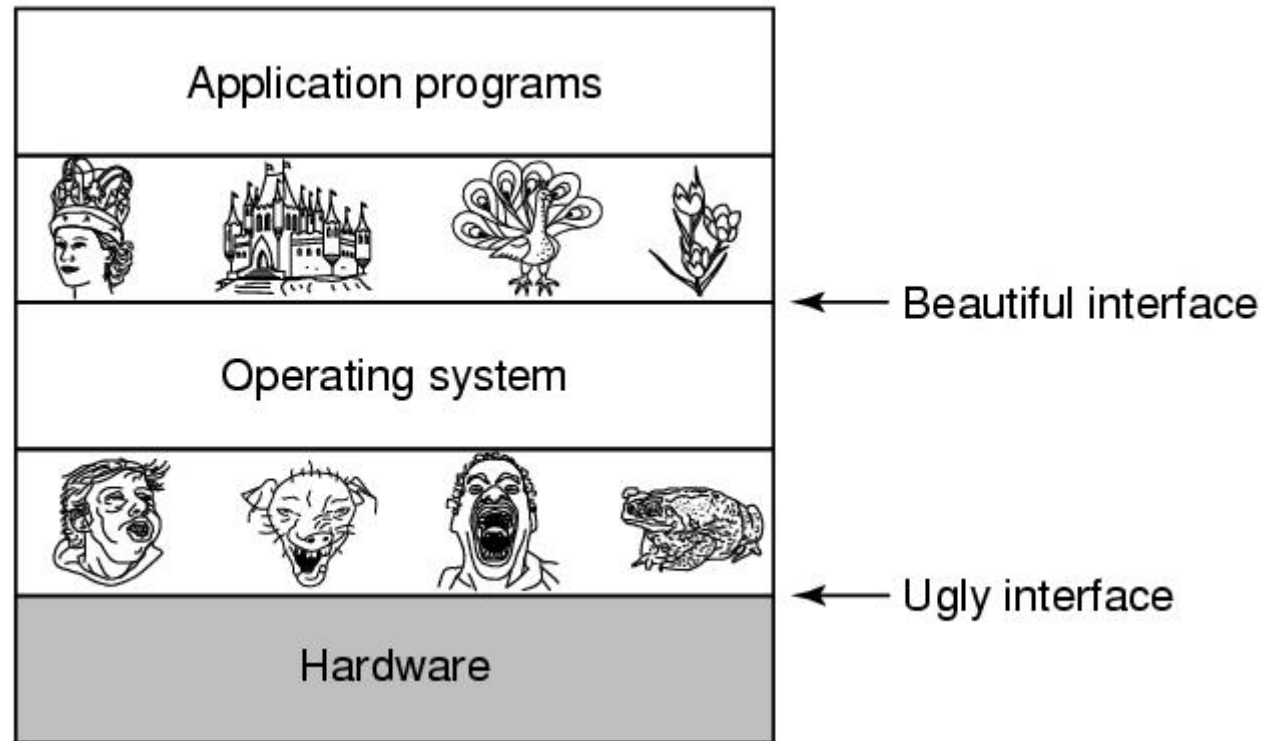
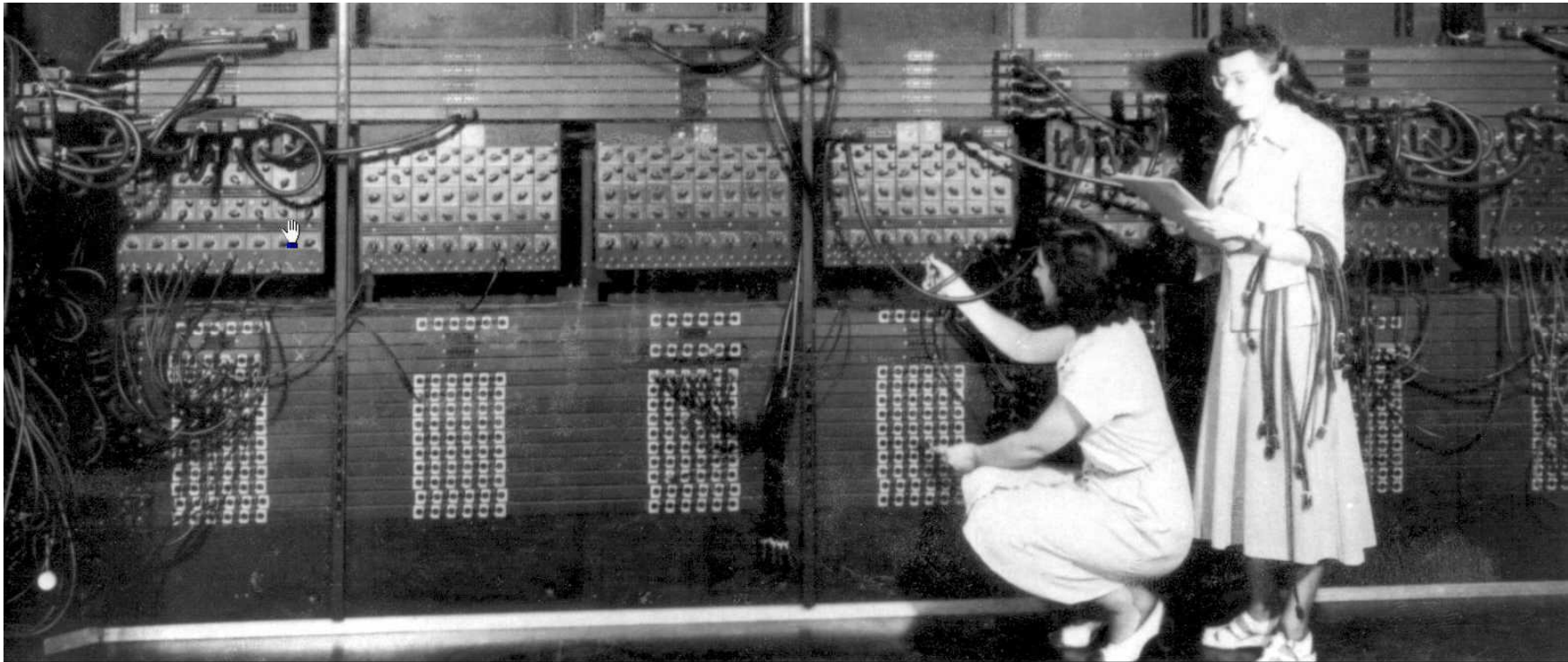


Figure 1-2. Operating systems turn ugly hardware into beautiful abstractions.

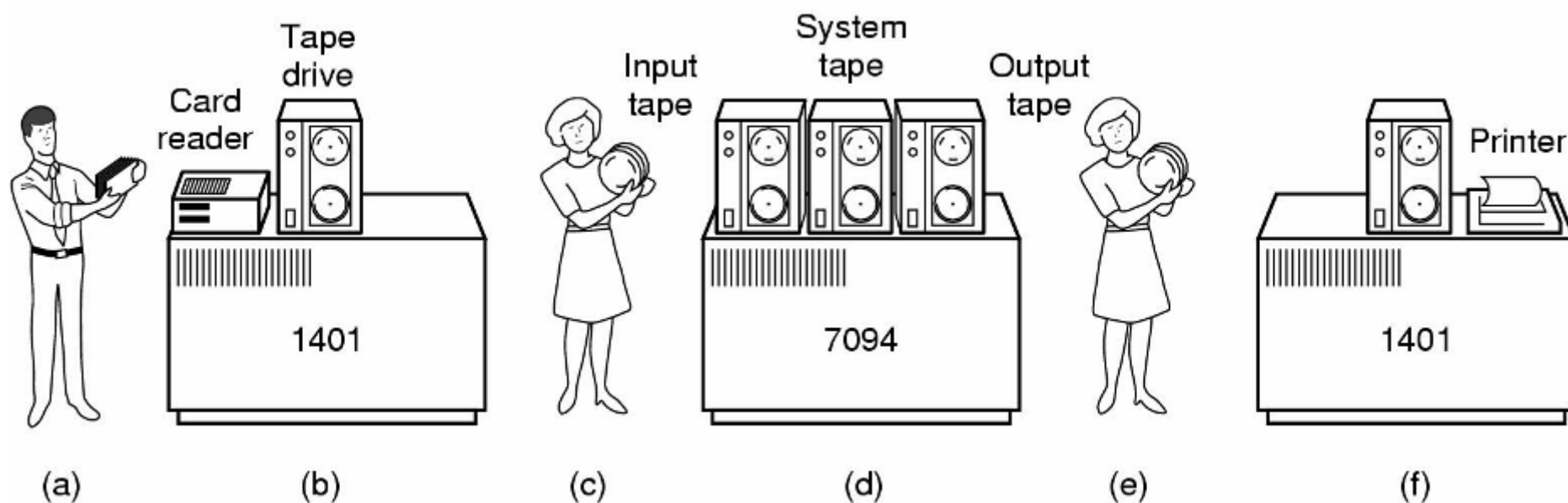
Историја на компјутерските системи

- ▶ Прва генерација 1945 – 1955
 - vacuum tubes, plug boards
- ▶ Втора генерација 1955 – 1965
 - transistors, batch systems
- ▶ Трета генерација 1965 – 1980
 - ICs and multiprogramming
- ▶ Четврта генерација 1980 – денешница
 - personal computers

Прва генерација (скап хардвер, ефтини програмери)

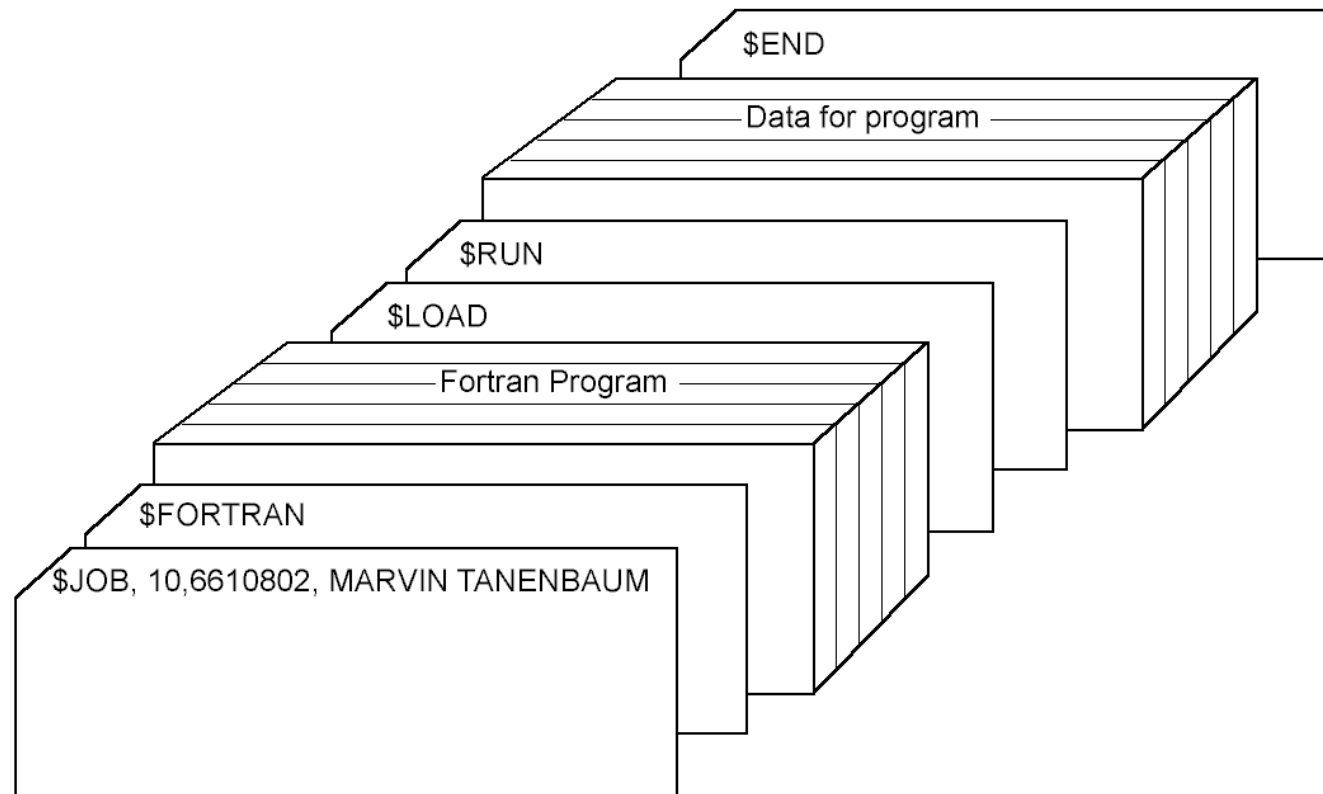


Втора генерација



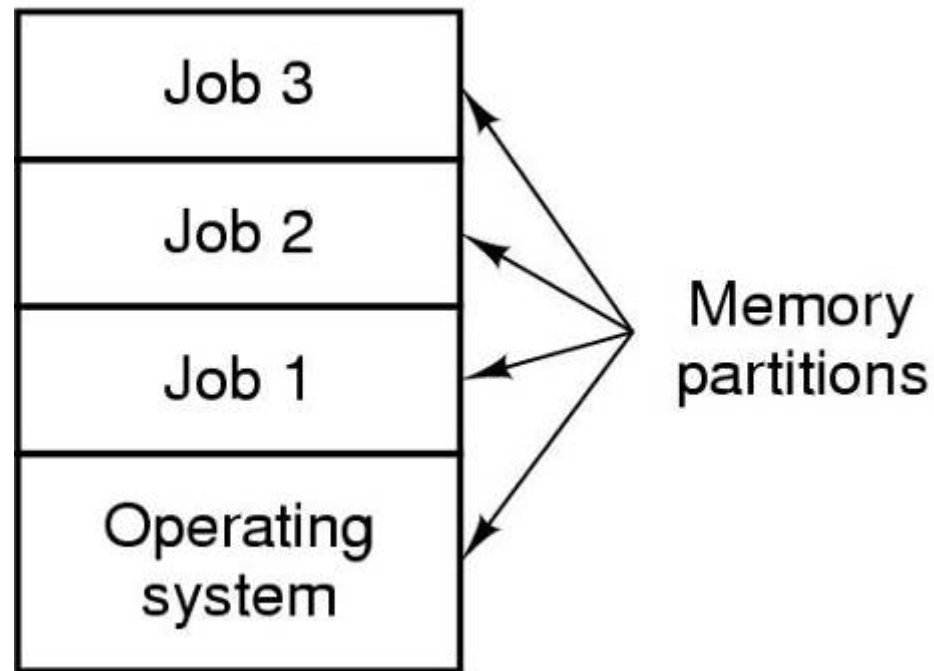
Раните batch системи

Втора генерација



Структура на типична работа кај 2-та генерација

Трета генерација (IC и Multiprogramming) – IBM System/360



- ▶ Мултипрограмирање (Multiprogramming)
 - Три jobs во меморијата

Трета генерација

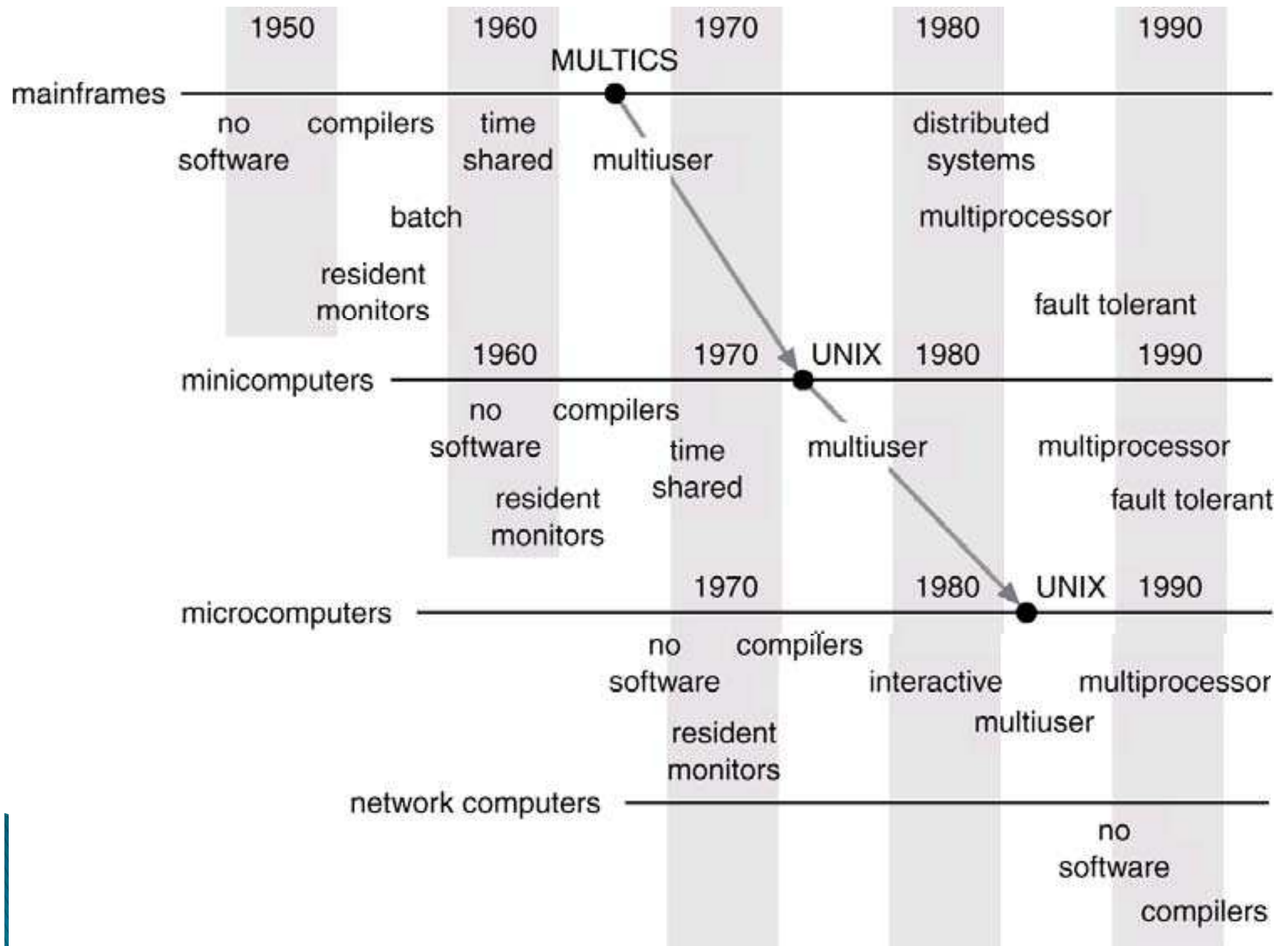
- ▶ Spooling (Simultaneous Peripheral Operation On Line)
- ▶ Тешко програмирање
- ▶ Timesharing – CTSS, MULTICS систем
- ▶ Minicomputers – серија PDP
- ▶ Корисничка верзија на MULTICS – UNIX
- ▶ Едукативна верзија – MINIX
- ▶ Продукциска верзија – LINUX

Четврта генерација

- ▶ Хардверот е ефтин, луѓето се скапи
- ▶ Персонални компјутери
- ▶ CP/M (Control Program for Microcomputers)
- ▶ MS-DOS
- ▶ Windows
- ▶ UNIX, LINUX
- ▶ GUI, мрежни оперативни системи, реално временски оперативни системи

Еволуција на особини

- ▶ Во исто време во кое особините на компјутерските системи се менуваат, се менуваат и особините на ОС
- ▶ РС– ата се здобиваат со посоефистициран хардвер и софтвер – се усложнуваат и ОС
- ▶ Особините кои порано биле достапни за скапите хардвери, сега стануваат достапни за сите кориснички системи



Концепти кај оперативните системи

- ▶ Процеси
- ▶ Ќорсокак (Deadlocks)
- ▶ Управување со меморија
- ▶ Влез/Излез
- ▶ Датотеки
- ▶ Сигурност

Процес != програма

- ▶ Процесот е една појава на програма што работи (се извршува)
- ▶ Различни процеси можат да работат, т.е. да бидат креирани од иста програма
- ▶ Во повеќето системи процесите формираат стебло на настанување чиј корен е првиот креиран процес
- ▶ Како минимум следниве ресурси се неопходни:
 - Меморија во која е содржан програмскиот код и податоците
 - Множество CPU регистри за поддршка на извршувањето

Аналогија

- ▶ Подготовка на роденденска торта. Постои рецепт и опремена кујна со сите потребни состојки: јајца, брашно, ореви...
- ▶ Рецептот (алгоритам зададен во соодветна нотација, разбирлива за куварите) – програма
- ▶ Куварот – CPU;
- ▶ Состојките – влезни податоци
- ▶ Активноста на куварот (читање на рецептот, додавање на состојките, печење на тортата) – ПРОЦЕСОТ

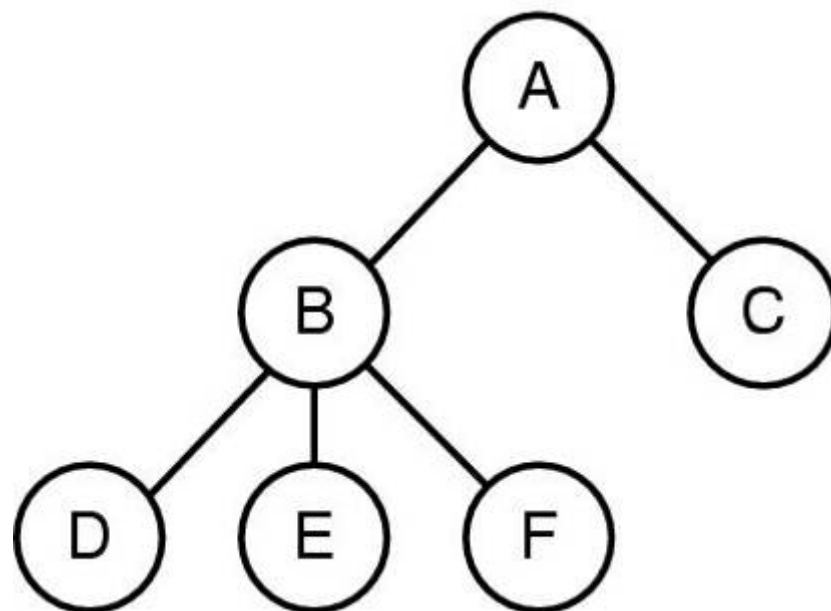
... Аналогија

- ▶ Прекин:
 - Помошникот на куварот се жали на повреда
- ▶ Куварот запамтува до каде бил со читање на рецептот (*состојбата на тековниот процес е зачувана*),
- ▶ зема книга за прва помош и започнува со следење на тие инструкции (*процесорот преминал од еден процес на друг, поприоритетен*).
- ▶ Секој од овие процеси има свој програм (рецепт за торта или книга за прва помош)
- ▶ Кога ќе ја санира повредата на својот помошник, куварот се враќа на правењето торта од таму каде што прекинал

Оттука

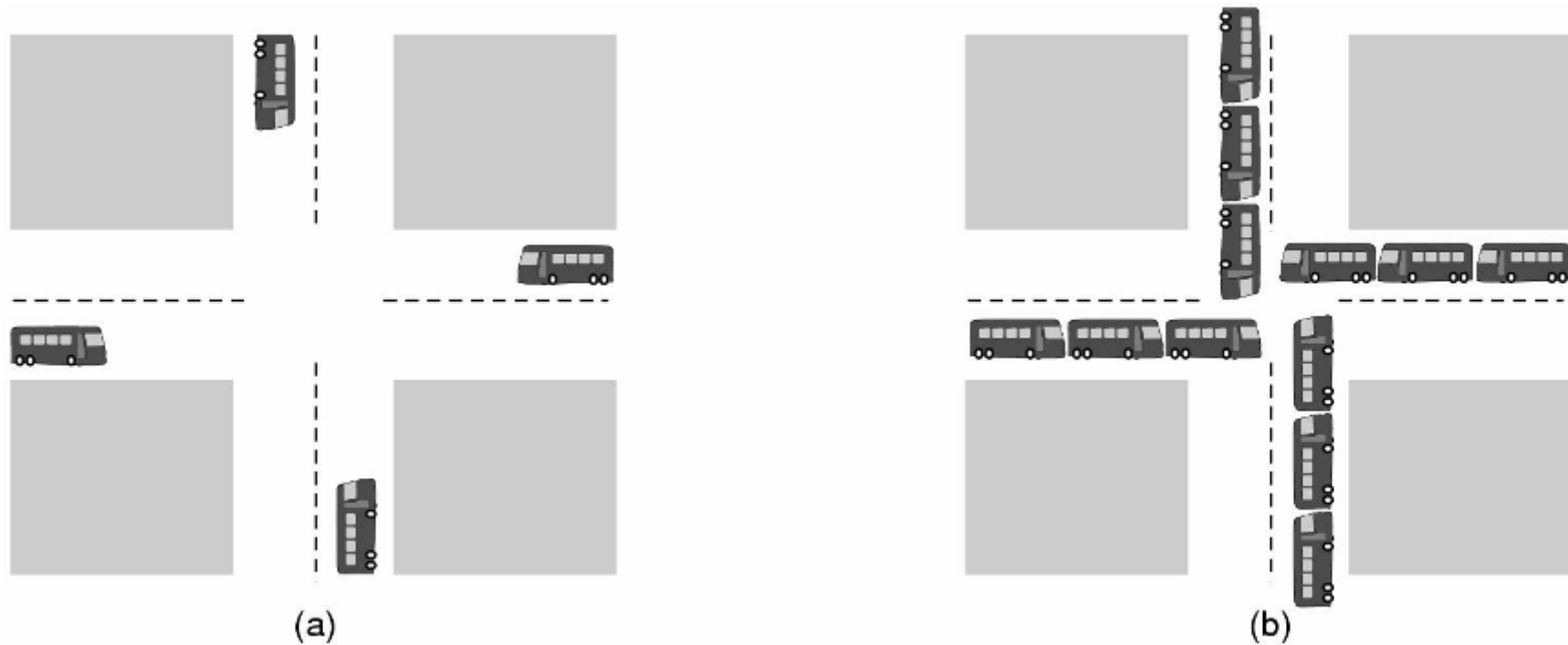
- ▶ Процес е некоја активност во извршување
- ▶ Тој има програма (код), влез, излез и состојби и ресурси кои ги поседува или дели.

Дрво на процеси



- ▶ Дрво од процеси
 - A ги креирал двата деца процеси, B и C
 - B ги креирал процесите, D, E, и F

Ќорсокак (deadlock)



(a) Потенцијален deadlock.
(b) deadlock.

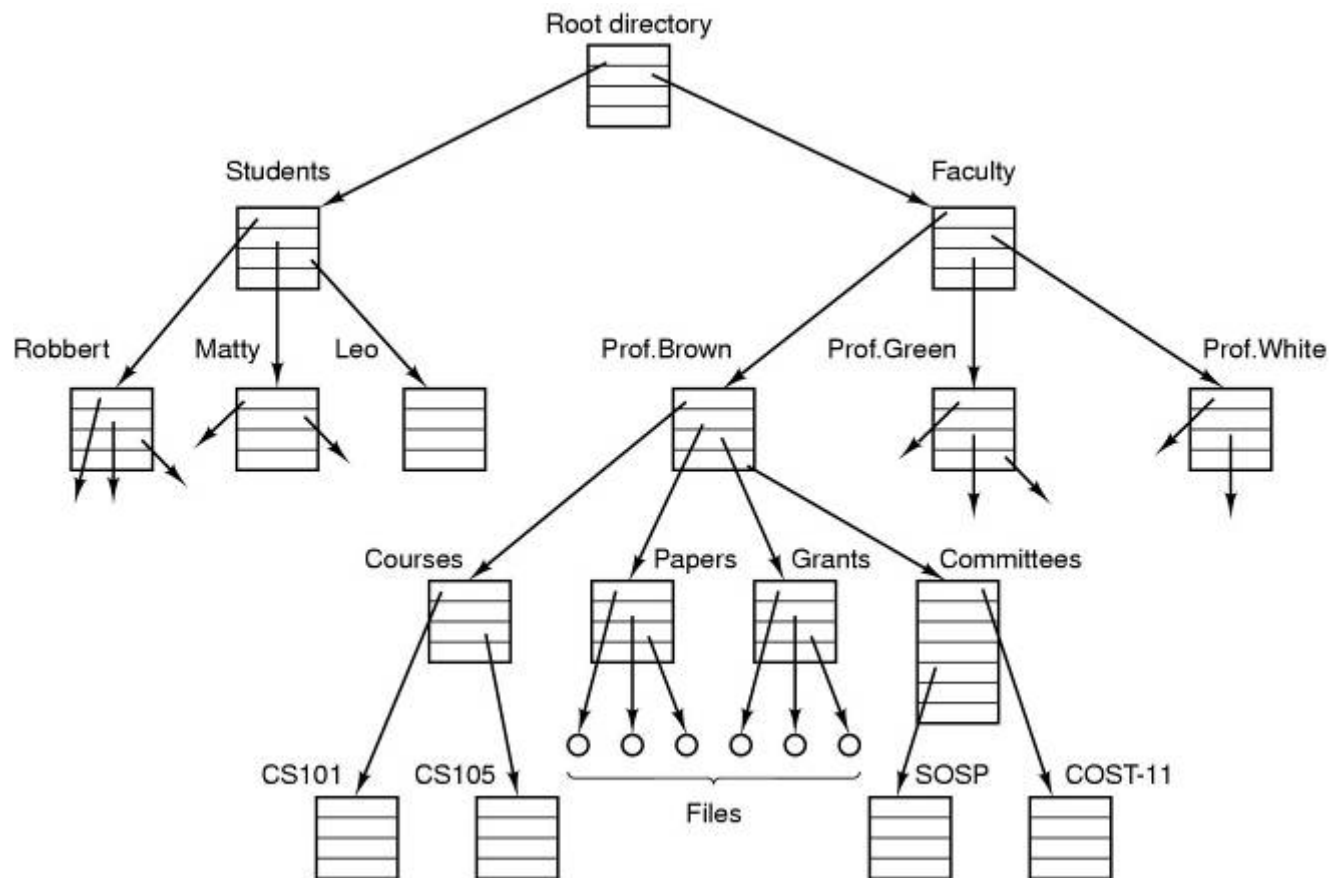
Управување со меморија

- ▶ раководење со меморијата
 - партиции
 - главната меморија се дели на неколку партиции — фиксна и динамичка поделба
 - разни алгоритми за сместување на процеси во меморија
- ▶ Виртуелна меморија

Датотечен систем

- ▶ Датотечниот систем е централен дел на секој оперативен систем.
- ▶ На корисниците им дава апстракција која многу ја поедноставува манипулацијата и управувањето со податоците на дискот – датотеките
- ▶ Мора да ги управува директно и уредите (дискот) и вклучува код за справување со грешки и ја обезбедува сигурноста на запишувањето (што кога нема струја, што има на дискот?), а има и детали на ниско ниво за достапување на податоците на дискот
- ▶ Најголемиот дел на датотечни системи се имплементирани во слоеви.

Систем од датотеки (File system)



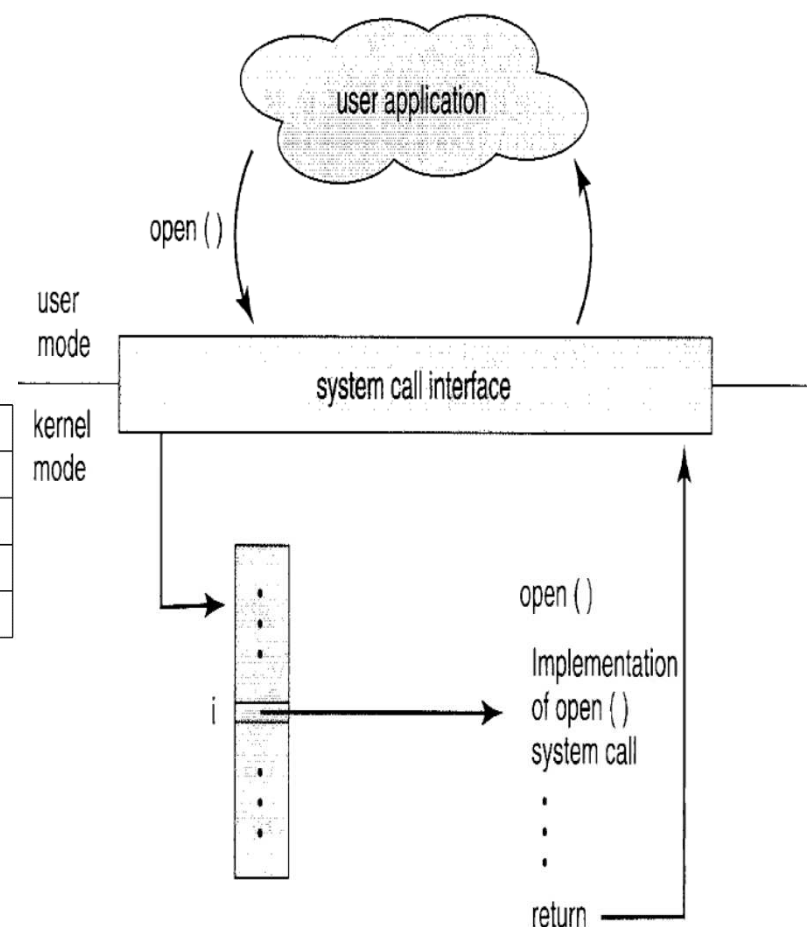
Пример на file system

Системски повици

- ▶ Системските повици претставуваат интерфејс за апликациите кон сервисите кои ги нуди ОС

Process management

Call	Description
<code>pid = fork()</code>	Create a child process identical to the parent
<code>pid = waitpid(pid, &statloc, options)</code>	Wait for a child to terminate
<code>s = execve(name, argv, environp)</code>	Replace a process' core image
<code>exit(status)</code>	Terminate process execution and return status

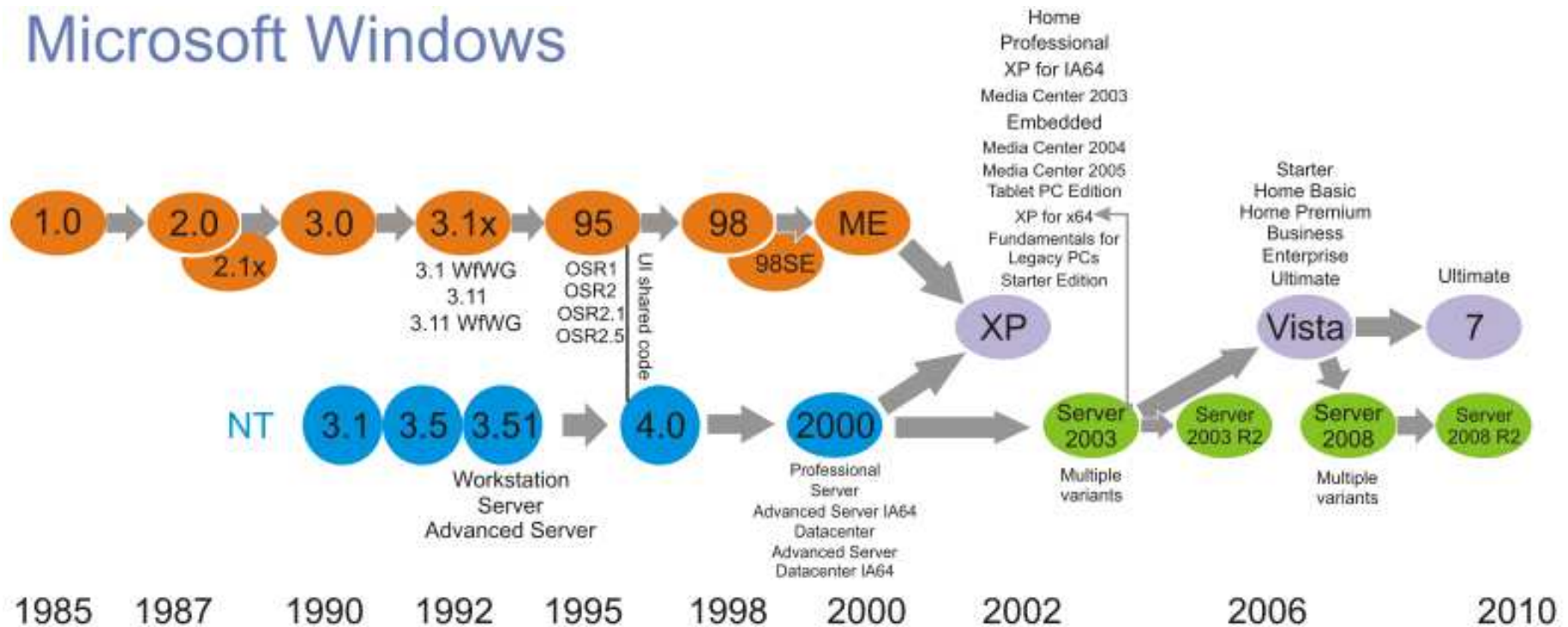


Зоолошка градина од ОС

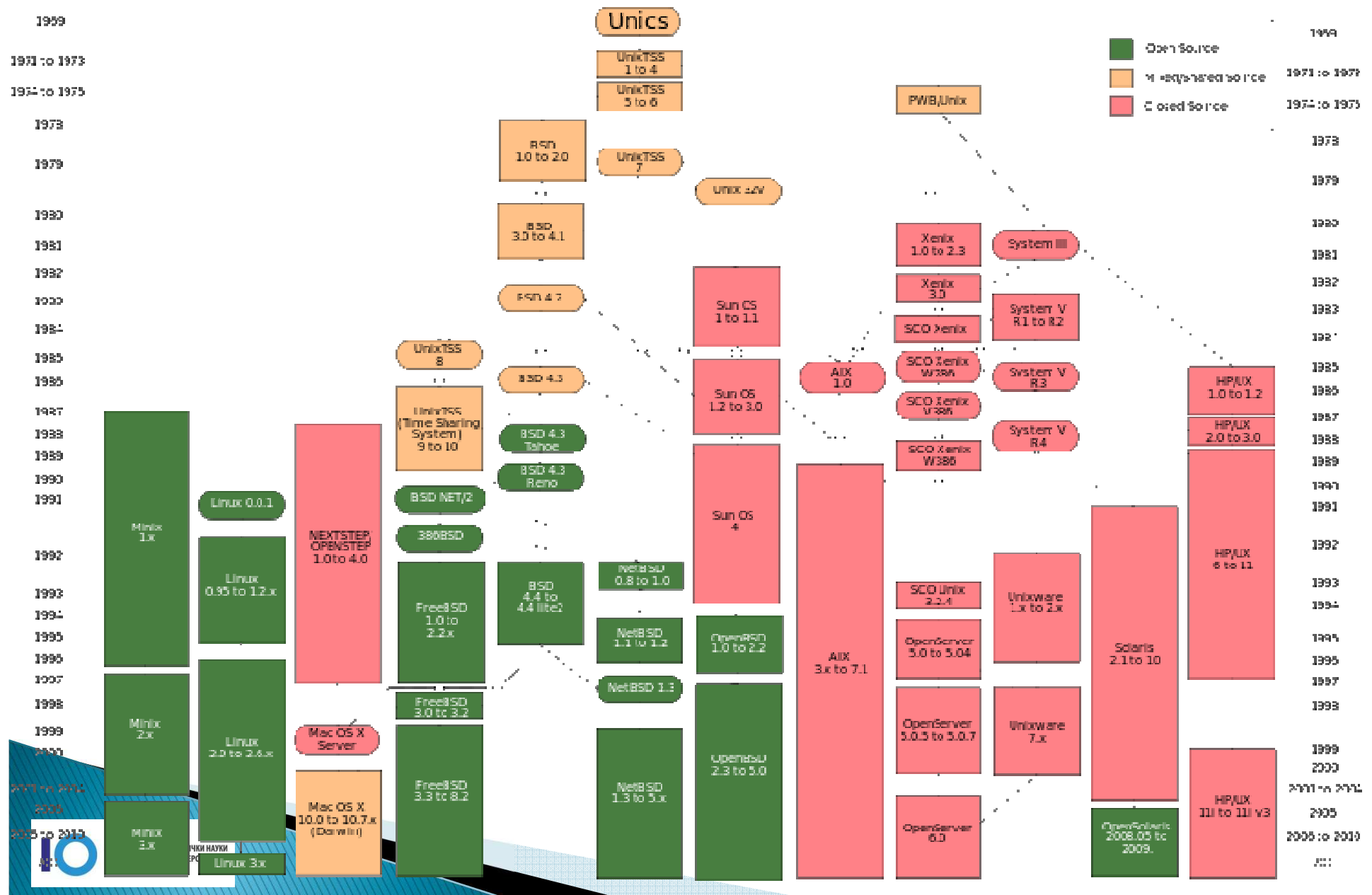
- Mainframe operating systems
- Server operating systems
- Multiprocessor operating systems
- Personal computer operating systems
- Mobile operating systems
- Embedded operating systems
- Sensor node operating systems
- Real-time operating systems
- Smart card operating systems

Windows operating systems

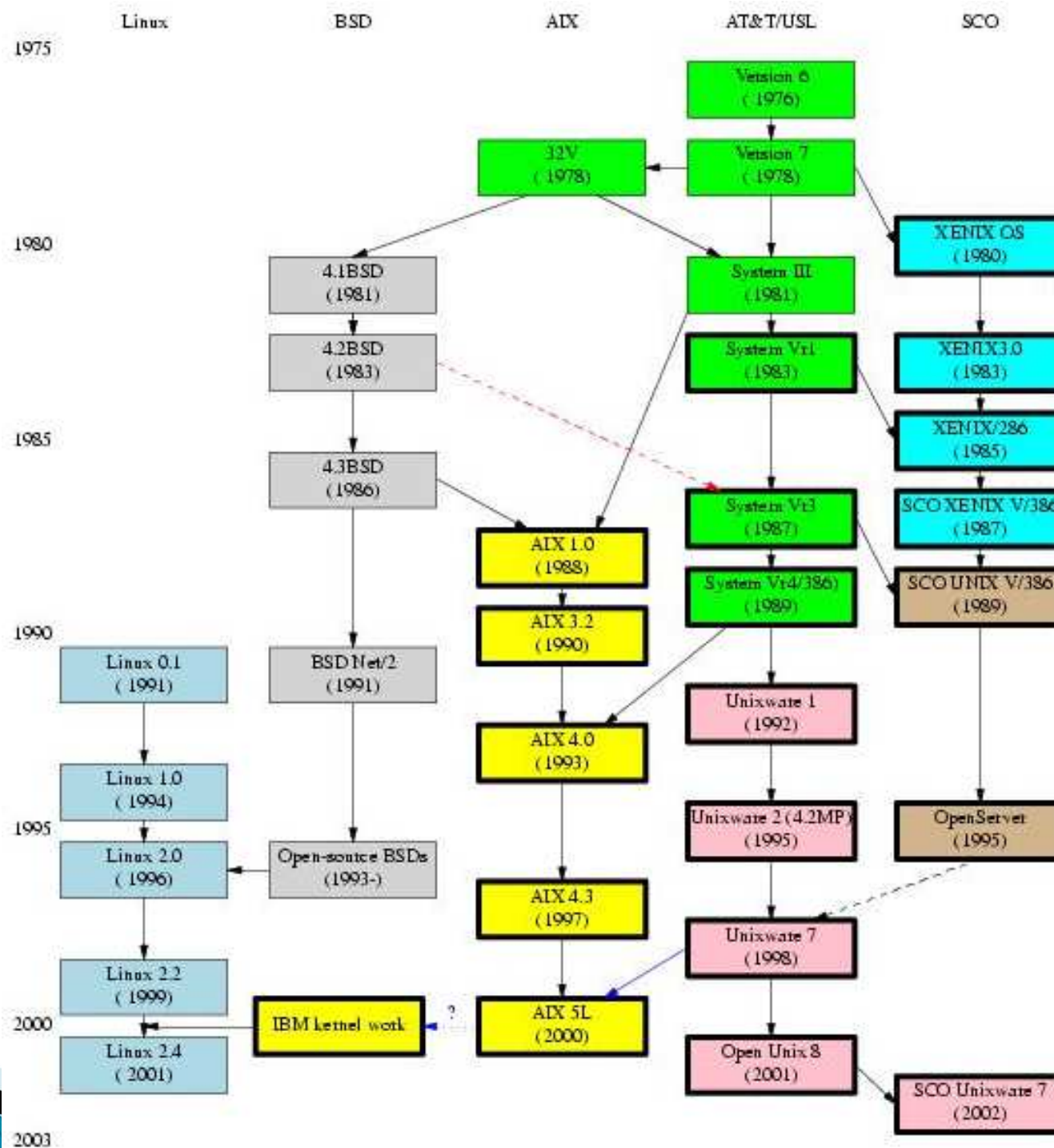
Microsoft Windows



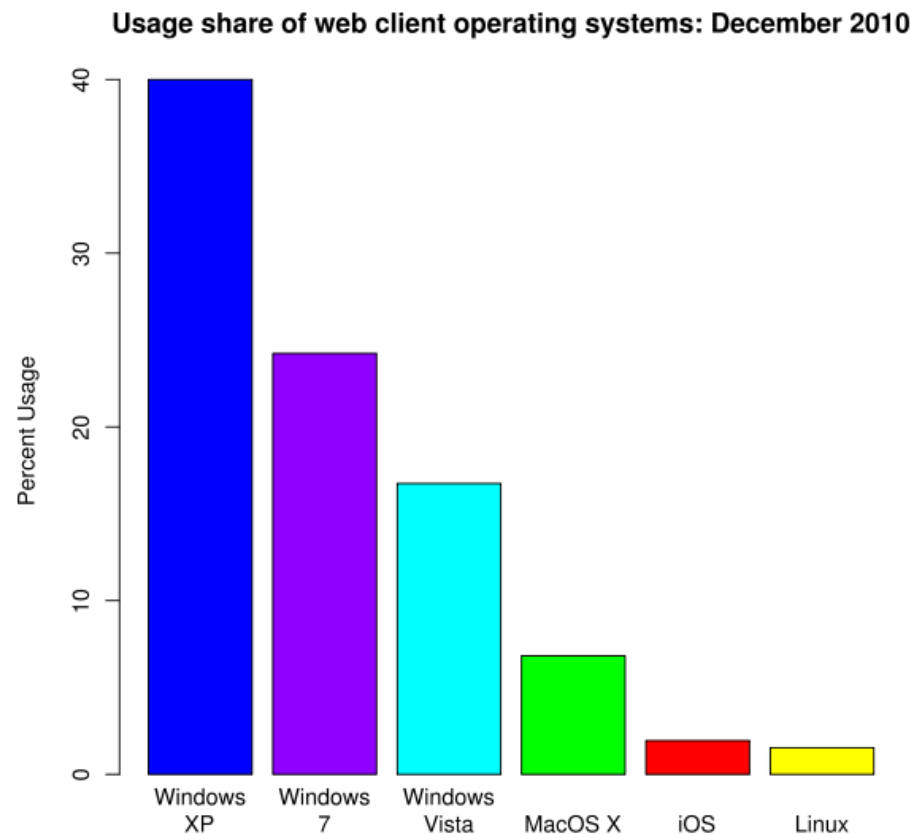
UNIX/Linux



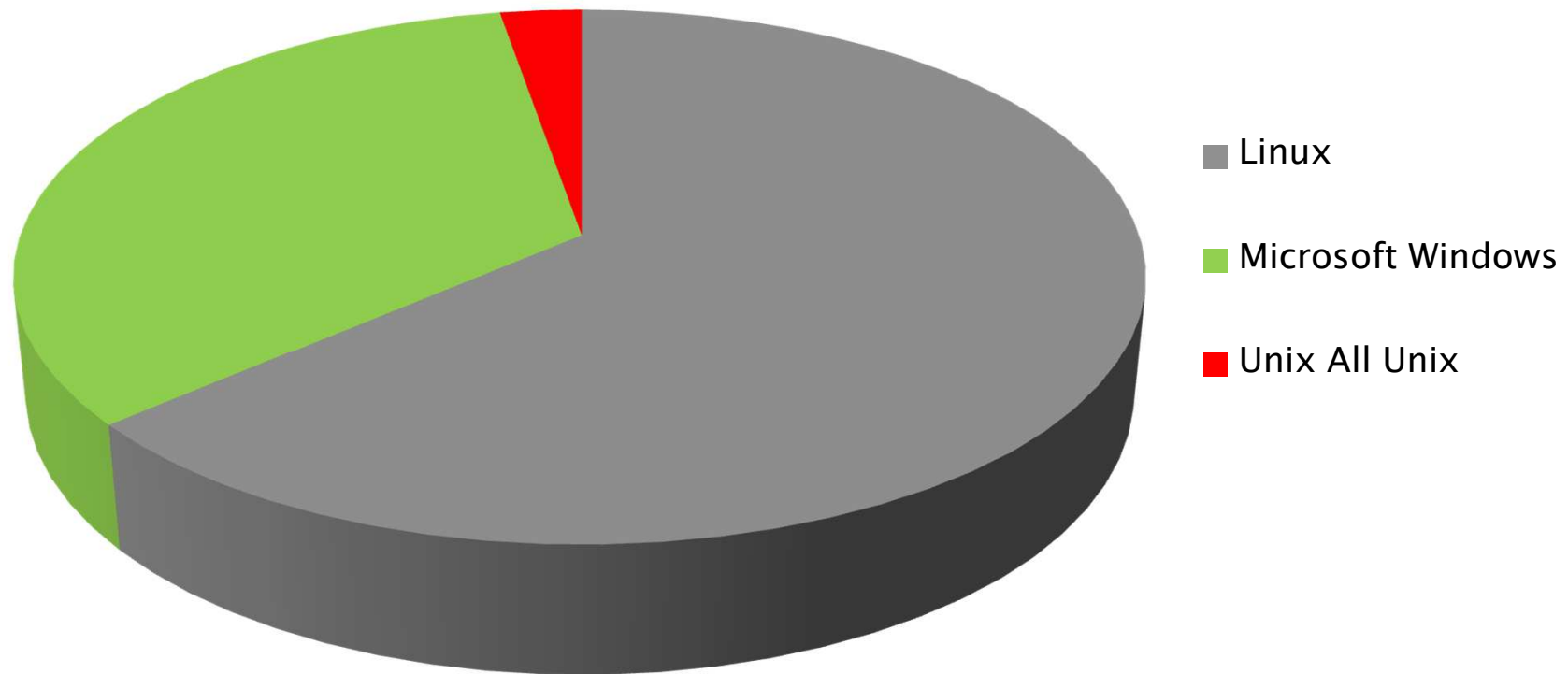
UNIX/Linux



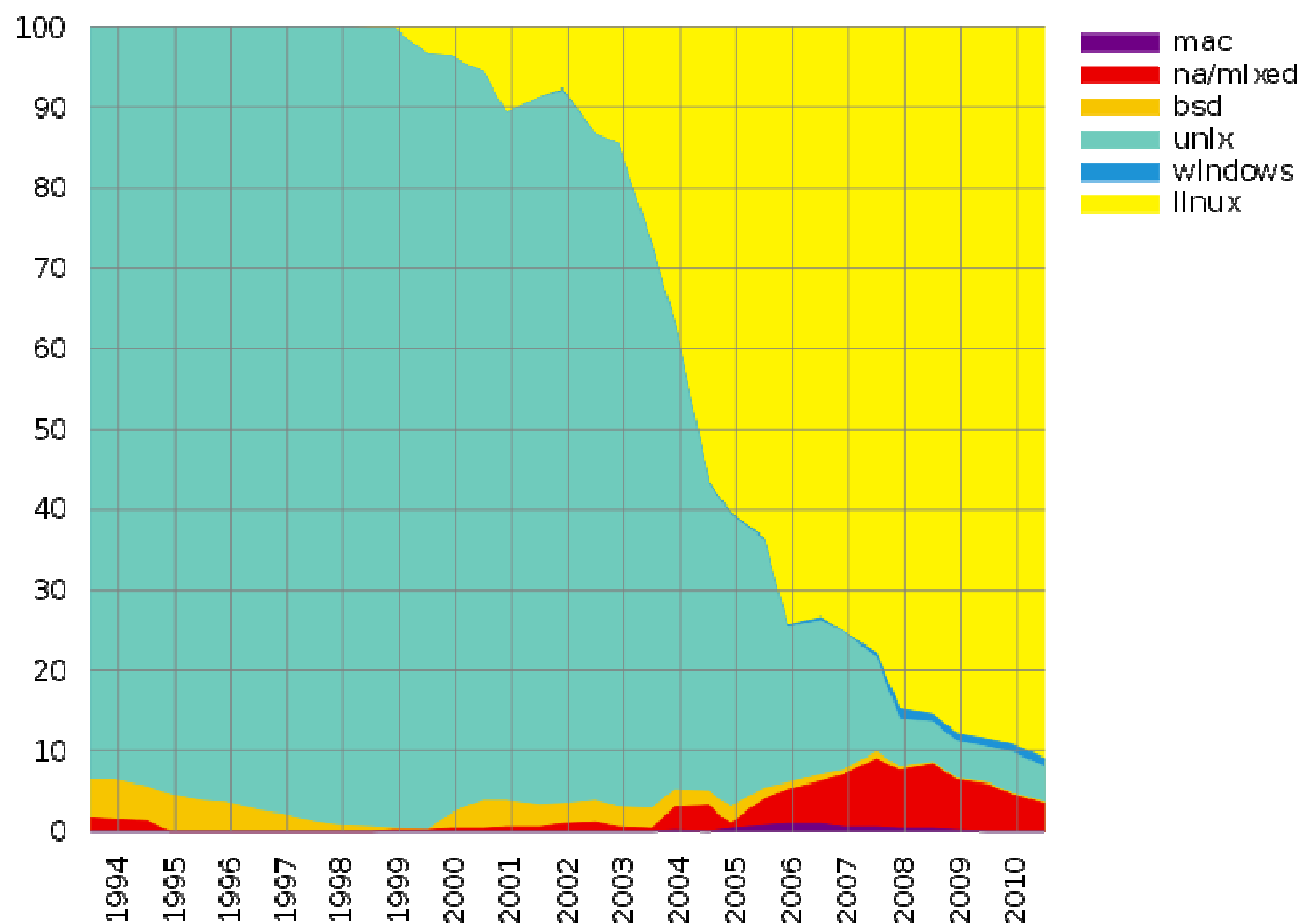
Застапеност на оперативните системи (web client)



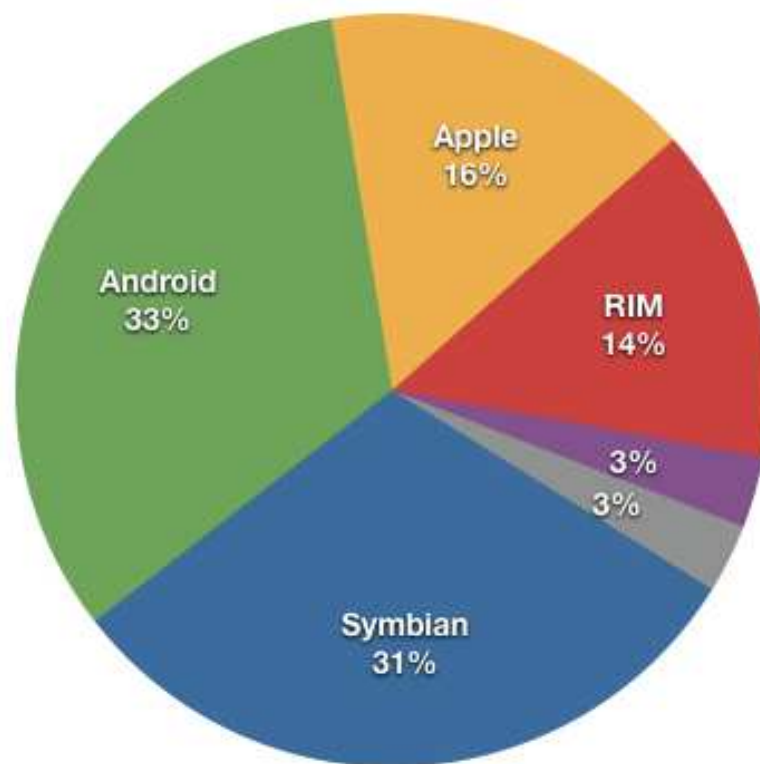
Застапеност на ОС сервери



Застапеност на ОС кај суперкомпјутерите



Застапеност во 2010, 2011 и 2012 Q3 на ОС кај паметните телефони според Gartner



● Symbian ● Android ● Apple ● RIM
● Windows Mobile ● Other

Operating System	3Q12 Units	3Q12 Market Share (%)	3Q11 Units	3Q11 Market Share (%)
Android	122,480.0	72.4	60,490.4	52.5
iOS	23,550.3	13.9	17,295.3	15.0
Research In Motion	8,946.8	5.3	12,701.1	11.0
Bada	5,054.7	3.0	2,478.5	2.2
Symbian	4,404.9	2.6	19,500.1	16.9
Microsoft	4,058.2	2.4	1,701.9	1.5
Others	683.7	0.4	1,018.1	0.9
Total	169,178.6	100.0	115,185.4	100.0

Заклучок

- ▶ Оперативниот систем обезбедува апстракција на виртуелна машина со цел полесно справување со различниот хардвер
- ▶ Оперативниот систем координира ресурси и ги штити корисниците еден од друг
- ▶ Оперативниот систем го поедноставува развојот на апликации преку обезбедување на стандарден интерфејс и апстракција
- ▶ Оперативниот систем може да обезбеди безбедност и толерантност од грешки како и опоравување по грешка