ОПЕРАЦІЙНІ СИСТЕМИ

- Цикл: Професійно-орієнтованих дисциплін
- Спеціальність 6.050103 «Програмна інженерія
- Автори: Качко Олена Григорівна, професор кафедри Програмної інженерії; Дягілєва Фаіна Григорівна, ст. викладач каф. Фізики НФаУ
- ekachko@gmail.com

ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДИСЦИПЛІНИ

Семестр	Лекції	Практичні заняття	Лабораторні роботи (годин/балів)	Самостійна робота
				Бали: Тест1/Тест2
4	36	12	24 (60)	20/20
				До 15.03 До 19.05

МЕТА ВИВЧЕННЯ

полягає у підготовці майбутніх спеціалістів для ефективного використання можливостей сучасних операційних систем

Знати:

- **класифікацію сучасних операційних систем;**
- **архітектуру та принципи побудови сучасних операційних систем;**
- **керування зовнішніми пристроями**;
- керування пам’яттю;
- забезпечення паралелізму в умовах одно та багато ядерних систем;
- керування процесами та потоками;

Вміти:

- **створювати та використовувати бібліотеки;**
- **використовувати функції операційних систем;**
- **використовувати найбільш ефективні засоби для роботи з об’єктами ядра операційної системи**
- використовувати різні засоби керування пам’яттю залежно від умов;
- створювати багато процесні та багато поточні програми з ефективним використанням усіх можливостей системи;
- виконувати синхронізацію доступу до загальних ресурсів з використанням найбільш придатних засобів, переключення між процесами та потоками.

ЛІТЕРАТУРА

(Google приблизно 10 млн. посилань)

- Основна література

1. Таненбаум. Э. Современные операционные системы. 3-е изд. –СПб.:Питер, 2010.-1120 с.
2. Бондаренко М.Ф., Качко О.Г. Операційні системи: навч. посібник. – Х.: Компанія СМІТ, 2008. – 432 с.

- Додаткова література

1 Рихтер Дж. Windows для профессионалов: создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows. - СПб.: Питер, 2001. – 752 с.

2 Рихтер Дж., Кларк Дж. Д. Программирование серверных приложений для Microsoft Windows 2000. СПб.: Питер, 2001. – 592 с.

ЛІТЕРАТУРА (ПРОДОВЖЕННЯ)

Навчальні посібники та наукові праці

- Бондаренко М.Ф., Качко Е.Г. Операционные системы. Учебное пособие. Х.: Компания СМІТ, 2006. – 402 с.
- Качко О.Г., Мельникова Р.В. Методичні вказівки до практичних занять з дисципліни „Операційні системи”. Харків, ХНУРЕ, 2009. – 88 с.
- Качко О.Г., Мельникова Р.В. Методичні вказівки до лабораторних занять. Харків: ХНУРЕ, 2009. – 44 с.

Електронні джерела

- **Сайти розробників процесорів** INTEL (**software.intel.com**, AMD (**developer.amd.com**), IBM (**ibm.com/developerworks**)
- **Сайти розробників операційних систем**: Microsoft (**http://msdn.microsoft.com/ru-ru/**), Linux (**http://www.linux.com/**), IBM (**http://www.ibm.com/developerworks/**), Hewlett-Packard (**https://developer.palm.com/**), Apple Computer (**http://developer.apple.com/**).
- **Сайты з науковими, учбовими і практичними матеріалами**: **http://www.intuit.ru-** Интернет университет Информационных технологий, winpedia.ru – энциклопедия по Windows,
- **Електронні журнали**: linux-mag.com, msdn.microsoft.com/en-us/magazine, ddj.com, pcmag.ru

РОЗДІЛИ КУРСУ ДЛЯ ВИВЧЕННЯ

Змістовий модуль 1 ОСНОВИ ОПЕРАЦІЙНИХ СИСТЕМ

Тема 1. ВСТУП. ОГЛЯД ОПЕРАЦІЙНИХ СИСТЕМ

Тема 2. ПОВТОРНЕ ВИКОРИТАННЯ КОДУ

Тема 3. ВВЕДЕННЯМ – ВИВЕДЕННЯ. СТАНДАРТНІ ТА НЕСТАНДАРТНІ ПРИСТРОЇ

Тема 4. ФАЙЛОВІ СИСТЕМИ

Тема 5 **КЕРУВАННЯ ПАМ’ЯТТЮ.** Фізична і віртуальна пам'ять

Тема 6 **КЕРУВАННЯ ПАМ’ЯТТЮ.** Кеш пам'ять

Тема 7 **КЕРУВАННЯ ПАМ’ЯТТЮ.** Динамічна пам'ять

Змістовий модуль 2. Процеси та потоки

Тема 1. АЛГОРИТМИ ТА ПОЛІТИКА ПЛАНУВАННЯ ТА ДИСПЕТЧЕРИЗАЦІЇ ПРОЦЕСІВ В УМОВАХ ОДНОЯДЕРНИХ ТА БАГАТОЯДЕРНИХ ПРОЦЕСОРІВ

Тема 2. Процеси . Створення та використання

Тема 3. Організація між процесорної взаємодії

Тема 4 Потоки. Створення та використання

Тема 5 Особливості синхронізації потоків

Тема 6. Асинхронне введення - виведення

ВСТУП. ІСТОРІЯ РОЗВИТКУ ОС, ФУНКЦІЇ

Питання для вивчення

- Вступ. Історія розвитку від найпростіших систем пакетної обробки до сучасних ОС.
- Функції сучасних ОС з погляду зручності роботи, ефективності та масштабування.
- Мережеві, клієнт - серверні, розподілені ОС та їх відмінності
- Проблеми безпеки ОС і програмних додатків та шляхи їх вирішення
- Вплив відкритого коду та Internet на розвиток сучасних ОС

ВСТУП. ІСТОРІЯ РОЗВИТКУ ВІД НАЙПРОСТІШИХ СИСТЕМ ПАКЕТНОЇ ОБРОБКИ ДО СУЧАСНИХ ОС

Основні визначення

- Системне програмне забезпечення (СПЗ)
- Системна програма (СП)
- Мови системного програмування (МСП)
- Операційна система (ОС)(комплекс програм для керування обчислювальною системою та додатками користувача з метою забезпечення найбільшої продуктивності цих додатків, їх розробників та користувачів

ІСТОРІЯ РОЗВИТКУ ОС

1 етап. Немає СПЗ

Єдина мова програмування – машинна (калькулятор)

Приклад програми для обчислення y = (a + b*x – c)/ (a - c) в машинних кодах

- 1 крок. Вивчення машинних кодів (+:01, -:02, *:03, /:04)
- 2 крок. Розподіл пам’яті:
- 3 крок. Складання програми
- 4 крок. Коректування програми

МАШИННА МОВА ПРОГРАМУВАННЯ. ПРОБЛЕМИ ВИКОРИСТАННЯ

Розподіл пам'яті		Програма					
Змінна	Адреса	Адреса	Код	A1	A2	A3	Comment
a	0	5	02	0	3	20	a-c
b	1	6	03	1	2	4	b*c
x	2	7	01	20	4	4	a+ b*c
c	3	8	04	4	20	4	y
y	4						

МАШИННА МОВА ПРОГРАМУВАННЯ. ПРОБЛЕМИ ВИКОРИСТАННЯ

Завдання на самостійну роботу.

Скласти машинну програму для обчислення $y=ax^3+bx^2+cx+d$.

Використати схему Горнера та схему з урахуванням можливості паралельного виконання

3 Порівняйте ці варіанти по кількості команд і допоміжної пам'яті.

4 запропонуйте найбільш ефективний варіант в разі, якщо процесор вміє виконувати операції паралельно.

ІСТОРІЯ РОЗВИТКУ ОС. БІБЛІОТЕКИ ТА ПАКЕТИ ПРИКЛАДНИХ ПРОГРАМ

Етап 1. Бібліотеки та пакети прикладних програм

- Мета – повторне використання коду

Типи бібліотек:

- Бібліотеки на мові програмування (бібліотеки класів), бібліотеки шаблонів;
- **Двійкові бібліотеки.**

Двійкові бібліотеки:

- Статичні та динамічні - використовуються при створенні додатку ↔виконанні додатку

Приклади пакетів: MathLab, Statistika,…

ОС:

Windows 7 приблизно 14000 бібліотек;

Windows 8 приблизно 34000 бібліотек;

ІСТОРІЯ РОЗВИТКУ ОС. БІБЛІОТЕКИ ТА ПАКЕТИ ПРИКЛАДНИХ ПРОГРАМ

Завдання на самостійну роботу.

Для середовища, яке ви використовуєте для створення програм, визначити типи бібліотек, які можна розробляти, і які використовуються самим середовищем.

ІСТОРІЯ РОЗВИТКУ ОС. СИСТЕМИ ПРОГРАМУВАННЯ

Етап 2. Системи програмування

Система програмування – мова програмування + транслятор+ компонувальник + завантажувач

Мови програмування:

- мови символічного кодування (асемблери)
- мови високого рівня (C, C++, C#, Java)
- мови для системного програмування

ІСТОРІЯ РОЗВИТКУ ОС. СИСТЕМИ ПРОГРАМУВАННЯ

Завдання на самостійну роботу

Які з визначених властивостей повинна мати мова системного програмування.

1 Створення найефективнішого коду.

2 Найпростіша для програміста.

3 Залежна від апаратури.

4 Мова має можливості керувати пам'яттю, доступ до показників, …

…

ІСТОРІЯ РОЗВИТКУ ОС. ПАКЕТНИЙ МОНІТОР

Призначення: Автоматизація виконання окремих компонентів завдання

Псевдокод

while (є програми в пакеті)
{
введення наступної програми;
трансляція, компоновка та завантаження програми
запуск програми
}

Проблеми:

1. Програма завершилась аварійно – не усі дані введені.
2. Якщо програма на початку пакета виконується довго, решта програм довго чекає.
3. Програма зациклилась – наступна програма не буде виконуватись.
4. Під час введення-виведення процесор чекає

Сучасні ОС:

командний файл – мова Shell, bat, cmd

Командний інтерпретатор cmd

ІСТОРІЯ РОЗВИТКУ ОС. СУЧАСНИЙ ПАКЕТНИЙ МОНІТОР

1 Виконання послідовності програм (калькулятор та WINWORD.EXE)
start %windir%\system32\calc.exe
"C:\Program Files\Microsoft Office\Office14\WINWORD.EXE "
2 Використання системних команд
Help > help.txt
Help copy > copy.txt
Help Delete > Delete.txt
3 Використання параметрів
%0 – ім'я командного файлу
%1 – ім'я 1 параметру, …, %9 – ім'я 9 параметру.

3.1 Приклад 1. Хай програма prog1.exe потребує задавати file1.txt в якості параметру, а prog2.exe – два файли file1.txt, file2.txt. Командний файл exec12.bat:

prog1.exe %1

prog2.exe %1 %2

Запуск командного файлу: exec2.bat x.txt y.txt

ІСТОРІЯ РОЗВИТКУ ОС. СУЧАСНИЙ ПАКЕТНИЙ МОНІТОР

Умовні оператори if < Умова> Оператор

1. **Відсутність параметру**

Приклад. Змінна кількість параметрів. Хай кількість параметрів 1. 2, 3.:

```
if ("%3") == ("") goto m2      echo %1 %2
echo %1 %2 %3                  goto mend
goto mend                      :m1
:m2                            echo %1
if ("%2") == ("") goto m1      :mend
```

2 ErrorLevel < Число >, not ErrorLevel < Число >

Хай prog1.exe виконує перетворення заданого файлу і результат перетворення записує в той же файл. Якщо при перетворенні виникла помилка, то заданий файл зіпсується. Як попередити це зіпсування і виконати запис результату тільки при успішному завершенні?.

```
Copy        %1 file.tmp /Y                  echo      prog1.exe: error
prog1.exe   %1                              copy      file.tmp %1 /Y
if          not ErrorLevel  1 goto ok       :ok
```

3 exist Ім’я файлу

not exist Ім’я файлу

Додати до попереднього командного файлу перевірку наявності файлу перед виконанням програми

If **not exist %1 goto error**

ІСТОРІЯ РОЗВИТКУ ОС. СУЧАСНИЙ ПАКЕТНИЙ МОНІТОР

Домашнє завдання

Написати командні файли для створення архіву файлів, які треба зберігати (Visual studio, C++), (Visual studio, C#). В якості параметру задавати: каталог для архіватору, ім'я solution, та імена додаткових проектів, якщо вони є.

1 Експериментально визначити, які файли та папки можуть бути видалені.

2 Написати команди для видалення визначених компонентів.

3 Виконати архівацію.

ІСТОРІЯ РОЗВИТКУ ОС. КОМАНДНІ ФАЙЛИ ТА VS

1 Використання команд операційної системи та командних файлів в середовищі VS

Properties→Build Events

Pre-Build Event

Command Line

Description

Use In Build

Pre-Link Event

…

Post-Build Event

…

Каталог для результату	$(SolutionDir)$(Configuration)\,
Ім'я проекту	$(ProjectName)
Робочий каталог	$(ProjectDir)
Файл результату	$(OutDir)$(TargetName)$(TargetExt)

ІСТОРІЯ РОЗВИТКУ ОС. КЕРУВАННЯ ВВЕДЕННЯМ – ВИВЕДЕННЯМ (I/0)

Алгоритм I/O для клавіатури

- Відстежити натиснення на клавішу.
- Відстежити відпускання клавіші.
- Перевірити, що в буфері клавіатури ще є місце.
- Перетворити номер обраної клавіші в код символу (Shift, CapsLock, Time, мова)
- Записати код (коди) символу в буфер.

Ці операції залежать від конкретного типу клавіатури, складно, (сотні, тисячі команд!!!)

ІСТОРІЯ РОЗВИТКУ ОС. КЕРУВАННЯ I/O

Контролер – програма та процесор для її виконання для керування I/O – платформено залежна – поставляється розробником пристрою. **Стандартний інтерфейс** (порти для вхідних і вихідних даних, статусу) – забезпечує незалежність від типу пристроїв.

Драйвер пристрою – працює безпосередньо з контролером – готовить дані для нього, та обробляє результати.

Драйвер пристрою + контролер – платформено залежна частина ОС (HAL – Hardware Abstract Level)

Драйвери логічних пристроїв – незалежні від конкретних пристроїв (дисків різних типів – флеш, жорсткий диск…, файлових систем)

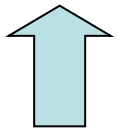
Обмін даними між драйверами різних рівнів - **Запрос I/O –** не залежить від конкретного пристрою.

IO Manager –> драйвери + механізми взаємодії між ними → синхронне + асинхронне введення – виведення.

ІСТОРІЯ РОЗВИТКУ ОС. КЕРУВАННЯ І/О БАГАТОПРОГРАМНИЙ РЕЖИМ

- Команди І/О винесені з загальної системи команд;
- Процесор не приймає участі в обробці запиту І/О;
- Синхронна та асинхронна обробка;
- Блокування програми. Запит та переривання
- Багатопрограмний режим без витіснення
- Режим розподілу часу – багатопрограмний режим з витісненням
- Накладні витрати, пов'язані з багатопрограмним режимом
- **Process Manager (**Планувальник + диспетчер програм**)**
- Системи реального часу.

ІСТОРІЯ РОЗВИТКУ ОС. РЕЖИМ БАГАТЬОХ КОРИСТУВАЧІВ



Потужність обчислювальних систем (ОС) + вартість

Одночасне використання декількома користувачами. Що треба вирішити:

1. Ідентифікація користувача (логін, пароль).
2. Розподіл ресурсів між користувачами (квоти на пам’ять, дисковий простір, процесорний час,…).
3. Забезпечення сумісного доступу до системних програм, файлів, баз даних.
4. Захист особистих даних користувача.

Process Manager + Security Manager

ІСТОРІЯ РОЗВИТКУ ОС. ОС ДЛЯ БАГАТОПРОЦЕСОРНИХ ТА БАГАТОЯДЕРНИХ СИСТЕМ

- Масштабування – рівномірний розподіл завантаження між окремими компонентами системи
- Розподіл даних між локальною пам'яттю для кожного з процесорів
- Організація зв'язку між процесорами (передача даних)
- Робота з загальною пам'яттю, когерентність, NUMA (Non-Uniform Memory Access), необхідність ексклюзивного доступу.

ІСТОРІЯ РОЗВИТКУ ОС. ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ

Безпека модулів та даних ОС

- Режим ядра (обмеження по даним та командам – апаратний рівень);
- Для усіх об'єктів ядра – атрибути безпеки: список доступу та режим успадкування дескрипторів. По замовченню – визначаються адміністратором. Перевищити неможливо.
- Цілісність драйверів;
- Попередження про використання небезпечних драйверів.

Безпека модулів і даних User

- Модуль – файл – атрибути безпеки;
- Використання EFS – при запису даних на диск вони автоматично шифруються, при читанні - розшифровуються
- Криптопровайдери (CSP) – бібліотека функцій для реалізації криптографічних алгоритмів.
- ЦСК для управління ключами відповідно PKI

Вимоги до цілком безпечної OS

- Теоретичний доказ правильності
- Усі користувачі, файли та програми виконують операції в самій системі, жодна операція не виконується за границями системи (OS Genesis)

ІСТОРІЯ РОЗВИТКУ ОС. КЕРУВАННЯ ПАМ'ЯТТЮ

Класи пам'яті: пространство I/0, фізична, віртуальна.

Memory manager

1 Виділення пам'яті для модулів ОС при її завантаженні – ось чому цей модуль завантажується першим.

2. Виділення/ визволення пам'яті для програм користувача

3 Висока швидкість знаходження блоку необхідного розміру (дефрагментація в пам'яті не допускається)

4. Забезпечення захисту пам'яті однієї програми від іншої (різний адресний простір)

5. Забезпечення можливості використання загальної пам'яті для різних програм.

ІСТОРІЯ РОЗВИТКУ ОС. 64 – БІТНІ ОС

- 64 бітні **обчислення** + 64 бітна **адресація** (16 экзабайт- 44 бита – **16 терабайт – иначе огромные таблицы страниц!!!**)

(2^{40}- tera 2^{50} – peta, 2^{60} – exa, 2^{70}-zetta, 2^{80}-yotta)

- Має сенс, якщо пам'яті більше ніж 4ГБ
- Швидше робота, якщо розмір файлу перевищує 2^{32};
- **Більше** пам'яті для додатків, які часто використовуються, завантажуються бібліотеки під час завантаження ОС (Windows 7)

ІСТОРІЯ РОЗВИТКУ ОС. 64 – БІТНІ ОС

- **Перша 64 –бітна ОС 1985: Cray Research випускає UNICOS** – (UNIX 64) для процесорів Cray-1 (**1976**)
- **2009 Microsoft + Windows 7**.
- **Обмеження по пам'яті**

ОС	Оперативна Ultimate/Home	Віртуальна
Windows 7	192GB/8GB	
Windows 8	512GB/128GB	
Windows 10	2TByte/128GB	7TByte

ІСТОРІЯ РОЗВИТКУ ОС. 64 – БІТНІ ОС + 64 – БІТНІ ДОДАТКИ

1. 32 бітні та 64 бітні апаратні драйвери не сумісні.
2. 2 режими роботи: long (64 біта), legacy (спадщина)
3. Підтримка 32 бітних додатків : WoW64 (Windows on Windows 64 для архітектури IA64 + Intel64)
4. Особливості створення 64 бітний додатків: (http://habrahabr.ru/company/intel/blog/93831/)
5. Адреси, показники, sizeof

ОСОБЛИВОСТІ WINDOWS8

Серпень 2012 – завершена розробка

1 При переставці – збереження файлів користувача

2 значно швидше завантаження ОС (ініціалізація пристроїв (20%), завантаження ядра (70%), ініціалізація даних користувача (10%)). При виході зберігається в файлі стан ядра, при повторному завантаженні – тільки відображення цього файлу (70%-20%)

3 захист –

W7: BIOS→Boot →Device Drives →Antivirus → OS Service

W8: UEFI→Boot $_{цп}$→Windows Core → Antivirus →Drives

UEFI (Unified Extensible Firmware Interface) – заміна BIOS, розроблено 140 компаніями,

4 Новий інтерфєйс – замість Пуск – Стіль Метро (як у IPad)

5 Одна версія ОС для комп'ютерів та планшетів – Одні додатки!

ОСОБЛИВОСТІ WINDOWS10

Липень 2015 (для ПК, ноутбуків, планшетів. Далі для мобільних телефонів)

1 Якщо у вас Windows 7, Windows 8 – безкоштовне оновлення.

2 Меню Пуск – об'єднані стиль Метро та робочий стіл. Не усі додатки та документи, а тільки, які використовуються найчастіше, можна змінювати розмір, об'єднувати в групи, …. Тут же кнопка off, настройки параметрів комп'ютера,

3 Вбудовано Cortana – голосовий помічник користувача з елементами штучного інтелекту, можна використовувати в якості перекладача, планування роботи, …

4 Новий Internet Explorer (Edge), поєднано з Cortana. (Старий Explorer теж залишено для сувмісності)

5 Віртуальний робочий стіл (є в MAC OS, Linux) – можна створити декілька зі своїм start menu, іконками,. Переключення між ними дуже просте.

6 **Windows Defender(антівірус, вбудований в Windows 8) – зручні засоби керування режимами роботи.**

7 Device Guard – можна встановити довірені драйвери, додатки підписати, решта додатків – блокуються.

8 Wi-Fi Sense – роботу з Wi-Fi потребують різні програми – Skype, Outlook, facebook, viber. Усім цим системам забезпечується робота. Достатньо настроїти для одного додатку.

9 Біометрична аутентифікація (скан обличчя, глаза, відбитки пальця). Потрібні додаткові пристрої для читання інформації.

10 Значно покращена система пошуку. Запити можна давати голосом. Для наступного запиту виводить попередній запит, не треба його повторювати.

ІСТОРІЯ РОЗВИТКУ ОС. ВИКОРИСТАННЯ МЕРЕЖ

Підтримка мереж (локальних, корпоративних, Internet) (дороге обладнання, Клієнт- серверні додатки, розподілені обчислення)

Хост + Канали передачі даних + Стандартний протокол обміну даними (TCP/IP)

- **Мережеві ОС**
- В 1980 році реалізована в UNIX

ІСТОРІЯ РОЗВИТКУ ОС. ІНТЕРФЕЙС МІЖ ОС ТА ДОДАТКАМИ КОРИСТУВАЧА

- **Виконання основних функцій** ОС OS API (Windows API) – MSDN, Ріхтер.
- **Звертання до драйверу ОС** – системний виклик (UNIX), IOCTL функції для Windows
- Проблема – різні виклики, програми не сумісні на рівні мови програмування

	UNIX	Windows
Process	fork	CreateProcess
File	create	CreateFile

ІСТОРІЯ РОЗВИТКУ ОС. СТАНДАРТИ ОС. POSIX

- **1969 рік** (Кен Томпсон + Денис Ритчи + Брайан Керниган)- **UNIX**
- **До 1970** року більш ніж 100 різних ОС
- **Portable Operating System Interface for Unix** (на рівні програми на мові програмування)
- **1988 г.** Міжнародна організація по стандартизації (ISO - International Organization for Standardization) разом з Міжнародною електротехнічною комісією (IEC - International Electrotechnical Commission) прийняли даний стандарт (POSIX) під назвою ISO/IEC 9945.

Що дає прийняття стандарту? Повторне використання коду.

Повна відповідність: сучасні UNIX подібні системи, MAC OS

В основному сумісні: Linux система

Частково сумісні: (можна встановити для останніх версій Windows (але це – не стандартна поставка)

ФУНКЦІЇ СУЧАСНИХ ОС

- Прийом команд або завдань на спеціальних мовах сценаріїв та їх виконання;
- Виділення пам'яті та завантаження програм та бібліотек для виконання;
- Виділення часу процесора для потоку програми;
- Розподіл пам'яті між компонентами ОС та додатками користувача , захист їх від взаємного впливу;
- Розподіл зовнішніх пристроїв між компонентами ОС та додатками користувача;
- Керування усіма фізичними пристроями з метою забезпечення максимальної продуктивності обчислювальної системи і (або) користувача;
- Планування та диспетчеризація задач з урахуванням заданої дисципліни обслуговування;
- Керування процесами та потоками в режимі квантування часу;
- Підтримка роботи з локальною, корпоративною мережами та INTERNET;
- Захист модулів та даних від несанкціонованого використання;
- Підтримка 64-бітних процесорів та додатків.

ВЛАСТИВОСТІ ОС

- **Продуктивність (**час завантаження і вивантаження ОС та додатків, швидкість файлових операцій**)**
- **Ефективність** (сама не потребує багато ресурсів, вимоги до обладнання)
- **Надійність та стійкість** (захист від помилок користувача, самовідновлення)
- **Гнучкість та розширюваність** (настройка в залежності від конкретної апаратури та вимог користувача)
- **Переносимість** (можливість використовувати апаратуру різних виробників)
- **Безпека** (захист даних та програм ОС та користувача, захист даних від несанкціонованого доступу). Мінімальні вимоги: для кожного ресурсу є власник і є права доступу.
- **Сумісність** (можливість виконання додатку, створеного в іншій ОС)
- **Зручність та ясність** (простота установки та використання кінцевим користувачем, потреба спеціальної підготовки)

ПОРІВНЯННЯ WINDOWS, LINUX, MAC OS

	Windows 10	MAC OS	LINUX
Продуктивність		+	++
Безпека	•Disk Encryption •Блокування додатків не з магазину та не підписанних	•Disk Encryption •Блокування додатків не з магазину та не підписанних. Може бути відмінено для конкретного додатку	++! Адміністратор не отримує повного доступу до user account
Віруси	•Defender malware scanner Починає працювати після деяких вірусних програм	•Xprotect – автоматичне блокування вірусних програм за рахунок Песочниці – неможливості зміни файлів в системній папці, які навіть не можуть побачити додатки користувача	якщо вірус потрапив в одну машину, він не може потрапити на іншу. Open Src - багато users Самих машин для LINUX значно менше
Аутентифікація	•Біометрична	•-	

СТАТИСТИКА ВИКОРИСТАННЯ ОС

Див.
http://hotlog.ru/global/os?month=1

• Відомості	12.2015
• Windows 7	43.40%
• Android	17%
• Windows XP	11%
• Windows 8	10.40%
• iPhone	5.14%
• Windows 10	4.25%
• iPad	3.11%
• MAC OS	1.72%
• Linux	1.15%

НАЙБІЛЬШ НЕБЕЗПЕЧНІ СЕРЕДОВИЩА РОЗРОБКИ

Див. https://info.veracode.com/state-of-software-security-report-volume6-pt2.html

- Classic ASP – 1 686 Баг/Мб (1 112 критических)
- ColdFusion – 262 Баг/Мб (227 критических)
- PHP – 184 Баг/Мб (47 критических)
- Java – 51 Баг/Мб (5,2 критических)
- .NET — 32 Баг/Мб (9,7 критических)
- C++ – 26 Баг/Мб (8,8 критических)
- iOS – 23 Баг/Мб (0,9 критических)
- Android – 11 Баг/Мб (0,4 критических)
- JavaScript — 8 Баг/Мб (0,9 критических)

ВИСНОВКИ

- Практично усі етапи розробки ОС втілені в сучасні системи (від бібліотек до багатопроцесорних систем).
- Сучасна ОС – дуже складна система, це видно по кількості і складності функцій, які вона виконує.
- Використання мови високого рівня С та рівня HAL, на якому виконуються усі апаратно – залежні функції, значно спрощує розробку ОС для нових обчислювальних систем.
- Використання функцій ОС при розробці додатків дозволяє створювати найбільш ефективні додатки.
- Мета вивчення цієї дисципліни – навчитися ефективно користуватися функціями сучасних ОС при експлуатації та розробці програмних систем.

ПИТАННЯ ДЛЯ САМОСТІЙНОГО ВИВЧЕННЯ

- Характеристика мови програмування С, як мови для створення операційних систем.
- Характеристика мови програмування асемблер, як мови для створення операційних систем.
- Використання переривань для керування пристроями.

МАТЕРІАЛИ ДЛЯ ЕКСПРЕС-КОНТРОЛЮ

- Чим відрізняється мова системного програмування від інших мов програмування.
- Які мови системного програмування Ви знаєте?
- Як в командному моніторі вирішити проблеми: зайвих даних, зациклювання, неефективного використання процесора
- Навіщо відстежувати відпускання клавіші для визначення коду символу (клавіатура)?
- Чим по призначенню відрізняються драйвери фізичних і драйвери логічних пристроїв.
- Функції IO Manager
- Чому команди I/O не входять до системи команд процесору?
- Чим відрізняються сінхронні та асінхронні операції I/O?
- З чим пов'язані накладні витрати для багатопрограмного режиму? Чи завжди вони є?
- Функції Process Manager
- Яким чином можна забезпечити сумісний доступ до одних файлів і ексклюзивний для інших?
- В якому разі необхідно забезпечити ексклюзивний доступ до даних в пам'яті?

МАТЕРІАЛИ ДЛЯ ЕКСПРЕС-КОНТРОЛЮ (ПРОДОВЖЕННЯ)

- Для чого більшість модулів ОС працюють в режимі ядра?
- Яким чином визначається порушення цілісності драйверів ОС?
- Які типи алгоритмів реалізують криптопровайдери. Для відповіді на це запитання подивіться допомогу про CSP. Що забезпечує використання цих алгоритмів?
- Як завантажується блок керування пам'яттю ОС?
- Яка пам’ять виділяється для програмного додатку при його запуску?
- За рахунок чого забезпечується захист адресного простору програми?
- Як задати загальну пам’ять для декількох програм?
- Порівняйте список параметрів для функцій create (Unix) , та CreateFile (Windows). Зробіть висновки відносно інформації про файли, яка задається. Що не співпадає?
- Чим відрізняються властивості ОС: продуктивність та ефективність, переносимість і сумісність