

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ
УКРАЇНИ**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ**

Кафедра кібербезпеки та DATA-технологій

Факультет №6

МЕТОДИЧНІ МАТЕРІАЛИ

ДО ПРАКТИЧНИХ ЗАНЯТЬ

з навчальної дисципліни

«Архітектура та структурно-логічні основи ЕОМ»

вибіркових компонент

**освітньої програми першого(бакалаврського) рівня вищої
освіти 125 «Кібербезпека та захист інформації»
(«Безпека інформаційних та комунікаційних систем»)**

м. Харків

2023 р.

ЗАТВЕРДЖЕНО

Науково-методичною радою
Харківського національного
університету внутрішніх справ
Протокол від 30.08.2023 № 7

²

СХВАЛЕНО

Вченою радою факультету №6
ХНУВС Протокол від 25.08.2023
№ 7

ПОГОДЖЕНО

Секцією науково-методичної ради
ХНУВС з технічних дисциплін
Протокол від 29.08.2023 № 7

Розглянуто на засіданні кафедри кібербезпеки та DATA-технологій
(протокол від 15.08.2023 № 8)

Розробники:

Доцент кафедри кібербезпеки та DATA-технологій ХНУВС, кандидат технічних наук, доцент Юрій ГОРЕЛОВ

Рецензенти:

*Доцент кафедри програмного забезпечення ХНУРЕ, к.т.н., доцент, Олексій Лановий;
Професор кафедри протидії кіберзлочинності факультету № 4 ХНУВС, к.т.н.,
доцент, Віталій Носов*

Розподіл часу навчальної дисципліни за темами (денна форма навчання)

Номер та найменування теми	Кількість годин відведених на вивчення навчальної дисципліни						Література, сторінки	Вид контролю
	Всього	з них:						
		Лекції	Семінарські заняття	Практичні заняття	Лабораторні заняття	Самостійна робота		
Тема № 1. Логіко-математичніоснови інформатики	24	10	–	10	–	8	[1]	
Тема № 2. Архітектура обчислювальних систем	22	6	4		-	10	[1,3,8,9,11]	
Тема № 3. Комп’ютерні мережі	20	4	4	-	-	10	[1,3,7,8,9]	
Тема № 4. Програмне забезпечення	24	4	4	2	-	14	[1,3,7,8,9]	
Всього за семестр №1	90	24	12	12	-	42		екзамен

Розподіл часу навчальної дисципліни темами (заочна та дистанційна форма навчання)

Номер та найменування теми	Кількість годин відведених на вивчення навчальної дисципліни						Література, сторінки	Вид контролю
	Всього	з них:						
		Лекції	Семінарські заняття	Практичні заняття	Лабораторні заняття	Самостійна робота		
Тема № 1. Логіко-математичні основи інформатики	20	2	4	–	–	20	[1]	
Тема № 2. Архітектура обчислювальних систем	26	2	–	2	-	20	[1,3,7, 9]	
Тема № 3. Комп'ютерні мережі	24	–	–	-	-	20	[1,3,7, 8,9]	
Тема № 4. Програмне забезпечення	20	4	4	2	-	20	[1,3,7, 8, 9]	
Всього за семестр №1	90	2	–	8	-	80		Екзамен

3.Методичні вказівки до практичних занять

Тема№1 Логіко-математичні основи інформатики

Практичне заняття: Таблиці істинності

Мета заняття: Вивчити методи побудови таблиць істинності для довільних булевих функцій

Час проведення: 2 год

Навчальні питання:

1. Булева змінні
2. Булеви функції
3. Таблиці істинності

1. Булеви змінні

Однією з форм задання булевих функцій є таблиці істинності, які містять усі можливі комбінації значень вхідних змінних та відповідні їм значення функції. Розглянемо таблиці істинності основних функцій двох змінних

Запереченням висловлювання x називається нове висловлювання, яке є істинним, якщо висловлювання x хибне, і хибним, якщо висловлювання x істинне. Заперечення висловлюванню x позначається і читається «не x » або «невірно, що x ».

Логічні значення висловлювання x можна описати за допомогою такої таблиці:

x	\bar{x}
0	1
1	0

Таблиці такого виду прийнято називати таблицями істинності.

Нехай x - висловлювання. Оскільки \bar{x} також є висловлюванням, то можна утворити заперечення висловленню \bar{x} , тобто висловлювання $\bar{\bar{x}}$, яке називається подвійним запереченням висловлюванню x .

Кон'юнкцією (логічним множенням) двох висловлювань x, y називається нове висловлювання, яке вважається істинним, якщо обидва висловлювання x, y істинні, і хибним, якщо принаймні одне з них помилкове.

Кон'юнкція висловлювань x, y позначається символом $x \& y$ або (xy) , читається « x і y ». Висловлювання x, y називаються членами кон'юнкції.

Логічні значення кон'юнкції описуються такою таблицею істинності:

x	y	$x \& y$
0	0	0
1	0	0
0	1	0
1	1	1

Наприклад, для висловлювань «6 ділиться на 2», «6 ділиться на 3» їх кон'юнкцією є вислів «6 ділиться на 2 і 6 ділиться на 3», який, очевидно, істинн.

З визначення операції кон'юнкції видно, що союз «і» в алгебрі логіки вживається в тому сенсі, що й у повсякденній мові. Але у звичайній мові не прийнято з'єднувати союзом «і» два висловлювання, далекі одне від одного за змістом, а в алгебрі логіки розглядається кон'юнкція двох будь-яких висловлювань.

2. Булеви функції

З визначення операції кон'юнкції і заперечення зрозуміло, що висловлювання $x \& \bar{x}$ завжди хибне.

Диз'юнкцією (логічним додаванням) двох висловлювань x, y називається нове висловлювання, яке вважається істинним, якщо принаймні одне з висловлювань x, y істинно, і хибним, якщо вони обидва хибні.

Диз'юнкція висловлювань x, y позначається символом $x \vee y$, читається « x або y ». Висловлювання x, y називаються членами диз'юнкції.

Логічні значення диз'юнкції описуються такою таблицею істинності:

x	y	$x \vee y$
1	1	1
1	0	1
0	1	1
0	0	0

0	0	0
---	---	---

Наприклад, вислів «У трикутнику DFE кут D або кут E гострий» істинно, оскільки обов'язково істинне принаймні одне з висловлювань: «У трикутнику DFE кут D гострий», «У трикутнику DFE кут E гострий».

В повсякденній мові союз «або» вживається в різному розумінні: виключає і не виключає. В алгебрі логіки союз «або» завжди вживається в розумінні не виключає.

Імплікацією двох висловлювань x , y називається нове висловлювання, яке вважається помилковим, якщо x істинне, а y - хибне, і істинним у всіх інших випадках.

Імплікація висловлювань x , y позначається символом $x \rightarrow y$, читається «якщо x , то y » або «із x випливає y ».

Висловлення x називається умовою або посилюючим, вислів y - наслідком або висновком, висловлювання $x \rightarrow y$ - проходженням або імплікацією.

Логічні значення операції імплікації описуються такою таблицею істинності:

x	y	$x \rightarrow y$
1	1	1
1	0	0
0	1	1
0	0	1

Наприклад, вислів «Якщо число 12 ділиться на 6, то воно ділиться на 3», очевидно, істинно, оскільки тут істинна посилка «Число 12 ділиться на 6» і істинний висновок «Число 12 ділиться на 3». «Якщо $2 \cdot 2 = 5$, то існує баба яга».

Вживання слів «якщо ..., то ...» в алгебрі логіки відрізняється від вживання їх у повсякденній мові, де, як правило, вважається, що, коли висловлювання x хибне, то висловлення «Якщо x , то y » взагалі не має сенсу. Крім того, будуючи пропозицію виду «якщо x , то y » у повсякденній мові, завжди маємо на увазі, що пропозиція y випливає з пропозиції x . Вживання слів «якщо ..., то ...» в математичній логіці не вимагає цього, оскільки в ній сенс висловлювань не розглядається.

Еквіваленцією (або еквівалентністю) двох висловлень x , y називається нове висловлювання, яке вважається істинним, коли обидва висловлювання x , y або одночасно істинні, або одночасно помилкові, і хибним у всіх інших випадках.

Еквіваленція висловлювань x , y позначається символом $x \leftrightarrow y$, читається «для того, щоб x , необхідно і достатньо, щоб y » або « x тоді і тільки тоді, коли y ». Висловлювання x , y називаються членами еквіваленції. Логічні значення операції еквіваленції описуються такою таблицею істинності:

x	y	$x \leftrightarrow y$
1	1	1
1	0	0
0	1	0
0	0	1

Наприклад, еквіваленція «Трикутник SPQ з вершиною S і основою PQ рівнобедрений тоді і тільки тоді, коли $SP = SQ$ » є істинною, оскільки висловлювання «Трикутник SPQ з вершиною S і основою PQ рівнобедрений» і «В трикутнику SPQ з вершиною S і основою PQ $SP = SQ$ » або одночасно істинні, або одночасно хибні.

Еквівалентність відіграє важливу роль у математичних доведеннях. Відомо, що значне число теорем формулюється у формі необхідних і достатніх умов, тобто у формі еквівалентності. В цьому випадку, знаючи про істинність або хибність одного з двох членів еквівалентності і довівши істинність самої еквівалентності, робиться висновок про істинність або хибність другого члена еквівалентності.

Стрілка Пірса записується так $x \downarrow y$, а читається як « x стрілка Пірса y ». Функція істина тоді і тільки тоді, коли хибні обидві її змінні:

x	y	$x \downarrow y$
1	1	0
1	0	0
0	1	0
0	0	1

Стрілка Пірса протилежна диз'юнкції і для неї справедливо:

$$x \downarrow y \equiv \overline{x \vee y} \equiv \overline{x} \wedge \overline{y}$$

(1.2)

Функція (штрих) Шеффера записується так $x|y$, а читається як « x штрих Шеффера y ». Функція помилкова тоді і тільки тоді, коли обидва значення змінних істинні:

x	y	$x y$
1	1	0
1	0	1
0	1	1
0	0	1

Функція Шеффера протилежна кон'юнкції і для неї справедливо:

$$x|y \equiv \overline{x \& y} \equiv \overline{x} \vee \overline{y}$$

$x_1=x$	1	1	0	0		
$x_0=y$	1	0	1	0	Позначення	Назва
0	0	0	0	0	$0=0$	тотожний нуль, детектор 0
1	0	0	0	1	$x \downarrow y$	стрілка Пірса, НЕ-АБО, 2АБО-НЕ, антидиз'юнкція, функція Даггера, функція Вебба, детектор 1
2	0	0	1	0	$x \leftarrow y$	інверсія зворотної імплікації, менше, детектор 2
3	0	0	1	1	$\neg x$	заперечення (негація, інверсія) першого операнда
4	0	1	0	0	$x \rightarrow y$	інверсія прямої імплікації, більше, детектор 4
5	0	1	0	1	$\neg y$	заперечення (негація, інверсія) другого операнда
6	0	1	1	0	$x \oplus y$	додавання по модулю 2, виключне «або», сума Жегалкіна, ^[1] не дорівнює
7	0	1	1	1	$x y$	штрих Шеффера, НЕ-І, 2І-НЕ, антикон'юнкція, пунктир Чулкова
8	1	0	0	0	$x \& y = x \cdot y$	кон'юнкція, 2І, мінімум, детектор 8
9	1	0	0	1	$(x \equiv y) = x \sim y = x \leftrightarrow y$	еквівалентність, рівність
10	1	0	1	0	y	другий операнд
11	1	0	1	1	$x \rightarrow y = x \supset y$	пряма (матеріальна) імплікація (від першого аргументу до другого), менше або дорівнює
12	1	1	0	0	x	перший операнд
13	1	1	0	1	$x \leftarrow y = x \subset y$	зворотня імплікація (від другого аргументу до першого), більше або дорівнює
14	1	1	1	0	$x \vee y = x + y$	диз'юнкція, 2АБО, максимум
15	1	1	1	1		тотожна одиниця, тавтологія

3. Таблиці істинності

Використовуючи розглянуті таблиці, можна побудувати таблицю істинності для будь-якої функції.

Наприклад, для формули $(x \sqcup y) \sqcup (x \& y)$ таблиця істинності така:

x	y	$(x \sqcup y)$	$x \& y$	$(x \sqcup y) \sqcup (x \& y)$
1	1	1	1	1
1	0	1	0	0
0	1	1	0	0
0	0	0	0	1

Приклад. Скласти таблицю істинності для формули $(X \rightarrow Y) \& (Y \rightarrow Z) \vee (Z \& X)$

Розв'язування.

- 1) Складемо таблицю.
- 2) Виконаємо логічну операцію - імплікації $(X \rightarrow Y)$.
- 3) Виконаємо логічну операцію - імплікації $(Y \rightarrow Z)$.
- 4) Виконаємо логічну операцію - кон'юнкцію $(X \rightarrow Y) \& (Y \rightarrow Z)$.
- 5) Виконаємо логічну операцію - кон'юнкцію $(Z \& X)$.
- 6) Виконаємо логічну операцію - диз'юнкцію $(X \rightarrow Y) \& (Y \rightarrow Z) \vee (Z \& X)$.

X	Y	Z	$X \rightarrow Y$	$Y \rightarrow Z$	$(X \rightarrow Y) \& (Y \rightarrow Z)$	$Z \& X$	$(X \rightarrow Y) \& (Y \rightarrow Z) \vee (Z \& X)$
0	0	0	1	1	1	0	1
0	0	1	1	1	1	0	1
0	1	0	1	0	0	0	0
0	1	1	1	1	1	0	1
1	0	0	0	1	0	0	0
1	0	1	0	1	0	1	1
1	1	0	1	0	0	0	0
1	1	1	1	1	1	1	1

Тема №1 Логіко-математичні основи інформатики

Практичне заняття: Спрощення логічних виразів

Мета заняття: Вивчити операції спрощення довільних булевих функцій

Час проведення: 2 год

Навчальні питання:

1. Основні властивості елементарних функцій
2. Спрощення формул

1. Основні властивості елементарних функцій

Логічні вирази можуть бути достатньо складними. Для їх спрощення використовуються тождества алгебри логіки та формули подання складних функцій через диз'юнкцію, кон'юнкцію та заперечення, тобто правила алгебра логіки. Основні представлені на рис.:

Основні властивості елементарних функцій:

- 1) Комутативність: $x \& y = y \& x$

$$x \vee y = y \vee x$$

$$x \vee y \neq y \vee x$$

- 2) Асоціативність: $(x \& y) \& z = (y \& z) \& x$

$$(x \vee y) \vee z \equiv (y \vee z) \vee x$$

$$(x \sqcap y) \sqcap z \equiv (y \sqcap z) \sqcap x$$

- 3) Дистрибутивність: $x \& (y \sqcup z) \equiv (x \& y) \sqcup (x \& z)$

$$x \sqcup (y \& z) \equiv (x \sqcup y) \& (x \sqcup z)$$

$$x \& (y \sqcup z) \equiv (x \& y) \sqcup (x \& z) (**)$$

- 4) Закони поглинання: $x \sqcup x \sqcap x; x \sqcup xy \sqcap x; x \sqcap xy \sqcup x \sqcap y$

$$x \& x \sqcap x \text{ (взагалі } x^n \sqcap x)$$

- 5) Закони нуля й одиниці $x \& 0 \sqcap 0; x \sqcap 0 \sqcap x; x \& 1 \sqcap x; x \sqcap 1 \sqcap 1; x \& x \sqcap 0; \bar{x} \sqcap x \sqcap 1$

- 6) Закон подвійного заперечення $x \sqcup \bar{\bar{x}}$

- 7) $x \sqcup y \sqcup x \sqcup y; x \sim y \sqcup xy \sqcup xy \sqcup x; x \sqcup y \sqcup xy \sqcup xy; x \sqcup y \sqcup x \sqcup y \sqcup x \& y;$
 $x / y \sqcup x \& y \sqcup x \sqcup y$

8) Закони Моргана $\overline{x \& y} = \overline{x} \vee \overline{y}$; $\overline{x \vee y} = \overline{x} \& \overline{y}$

Необхідно спростити вирази, використовуючи зазначені правила.

Тема №1 Логіко-математичні основи інформатики

Практичне заняття: Мінімізація ДНФ та КНФ

Мета заняття: Вивчити операції мінімізації ДНФ та КНФ

Час проведення: 2 год

Навчальні питання:

1. ДНФ та КНФ

2. МДНФ та МКНФ

1. ДНФ та КНФ

Диз'юнктивна нормальна форма (ДНФ) - це сума добутків, утворених зі змінних і їх заперечень. Диз'юнктивна нормальна форма не містить дужок.

Наприклад, форма $x \vee \overline{y} \& z$ - диз'юнктивна нормальна форма, а $x \& (x \vee \overline{y})$ - ні.

Кон'юнктивна нормальна форма (КНФ) - це добуток сум, що складаються зі змінних і їх заперечень.

Наприклад, форми $(a+b)c$, $ab(c+b)$, \bar{c} , a - кон'юнктивні нормальні форми.

Теорема. Всяка складна логічна функція може бути зведеною до диз'юнктивної нормальної форми.

Аби зробити це, необхідно:

- записати булеву функцію використовуючи операції $+$, \cdot , $-$;
- за допомогою законів де Моргана опустити риску заперечення до окремих літер і за законом подвійного заперечення знищити подвійні риси;
- за допомогою першого закону дистрибутивності знищити всі добутки сум і провести поглинання.

Одержана форма задовольняє визначенню диз'юнктивної нормальної форми.

Якщо диз'юнктивна нормальна форма функції від n змінних у кожній своїй кон'юнкції містить всі n змінних або їх заперечення, то це *досконала диз'юнктивна нормальна форма (ДДНФ)*. Кожна функція має одну-єдину досконала диз'юнктивну нормальну форму і вона може бути одержаною з таблиці істинності цієї функції шляхом запису через знак логічного додавання всіх наборів змінних, на яких ця функція визначена як істинна. Кожний такий набір змінних відповідає кон'юнкції.

Наприклад, для функції $F = (x_1 \vee x_2) \vee x_3$

досконала диз'юнктивна нормальна форма за таблицею істинності може бути побудованою так:

$x_1 x_2 x_3$	$F = (x_1 \vee x_2) \vee x_3$
0 0 0	1
0 0 1	0
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	0

$$\begin{aligned} \text{ДДНФ}(F) &= 000 + 010 + 100 + 101 + 110 = \\ &= \overline{x_1} \overline{x_2} \overline{x_3} \vee \overline{x_1} x_2 \overline{x_3} \vee x_1 \overline{x_2} \overline{x_3} \vee x_1 \overline{x_2} x_3 \vee x_1 x_2 \overline{x_3} \end{aligned}$$

Теорема. Всяка складна логічна функція може бути зведеною до кон'юнктивної нормальної форми.

Аби зробити це, необхідно:

- записати булеву функцію використовуючи операції $+$, \cdot , $-$;
- за допомогою законів де Моргана опустити риску заперечення до окремих літер і за законом подвійного заперечення знищити подвійні рисочки;
- за допомогою другого закону дистрибутивності знищити всі суми добутків і провести поглинання.

Одержана форма задовольняє визначенню кон'юнктивної нормальної форми.

Якщо кон'юнктивна нормальна форма функції від n змінних у кожній своїй кон'юнкції містить всі n змінних або їх заперечення, то це досконала *кон'юнктивна нормальна форма*. Кожна функція має одну-єдину досконала кон'юнктивну нормальну форму і вона може бути одержаною з таблиці істинності цієї функції шляхом запису через знак логічного множення всіх наборів змінних, на яких ця функція визначена як помилкова.

Наприклад, для функції $F = (x_1 \vee x_2) \vee x_3$ досконала кон'юнктивна нормальна форма за

таблицею істинності може бути побудованою так:

$x_1 x_2 x_3$	$F \square (x_1 \square x_2) \square \overline{x_3}$
0 0 0	1
0 0 1	0
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	0

ДКНФ=

$001 \square 011 \square 111 \square (x_1 \square x_2 \square x_3) \square (x_1 \square x_2 \square x_3) \square (x_1 \square x_2 \square x_3)$.

Приклад. Скласти таблицю істинності для формули $(X \rightarrow Y) \& (Y \rightarrow Z) \vee (Z \& X)$
Розв'язування.

- 1) Складемо таблицю.
- 2) Виконаємо логічну операцію - імплікації $(X \rightarrow Y)$.
- 3) Виконаємо логічну операцію - імплікації $(Y \rightarrow Z)$.
- 4) Виконаємо логічну операцію - кон'юнкцію $(X \rightarrow Y) \& (Y \rightarrow Z)$.
- 5) Виконаємо логічну операцію - кон'юнкцію $(Z \& X)$.
- 6) Виконаємо логічну операцію - диз'юнкцію $(X \rightarrow Y) \& (Y \rightarrow Z) \vee (Z \& X)$.

X	Y	Z	$X \rightarrow Y$	$Y \rightarrow Z$	$(X \rightarrow Y) \& (Y \rightarrow Z)$	$Z \& X$	$(X \rightarrow Y) \& (Y \rightarrow Z) \vee (Z \& X)$
0	0	0	1	1	1	0	1
0	0	1	1	1	1	0	1
0	1	0	1	0	0	0	0
0	1	1	1	1	1	0	1
1	0	0	0	1	0	0	0
1	0	1	0	1	0	1	1
1	1	0	1	0	0	0	0
1	1	1	1	1	1	1	1

2 МДНФ та МКНФ

Одна із основних задач є мінімізація ДНФ. Чим простіший логічний вираз, тим простіша і дешевша комбінаційна схема, що реалізує його.

Спрощення ДНФ можна здійснювати за допомогою законів та тотожностей алгебри логіки. Такий шлях мінімізації трудомісткий, тому в більшості випадків використовують графічні методи мінімізації (діаграми Вейча, діаграми Венна, карти Карно). Перевагою таких методів є наочність і простота використання при невеликій кількості змінних ($n \leq 4$).

Карти Карно. В 1953 році Моріс Карно опублікував статтю про розроблену ним систему графічного подання і спрощення ДНФ. Карта Карно показана на рисунку 1.2. Тут і надалі приймемо позначення $f(v) = y$

Входи		Вихід
x_2	x_1	Y
0	0	
1	0	
0	1	
1	1	

$\overline{x_2}$	x_2
$\overline{x_1}$	1
x_1	2
	3
	4

Рисунок 1.2 - Позначення квадратів на карті Карно

Чотири квадрати (1, 2, 3, 4) відповідають чотирьом можливим комбінаціям x_1 і x_2 в таблиці істинності функції з двома змінними. При такому зображенні квадрат 1 відповідає добутку $\overline{x_1} \cdot \overline{x_2}$, квадрат 2 - $\overline{x_1} \cdot x_2$ і т.д.

Припустимо тепер, що потрібно скласти карту Карно для функції, записаній у ДДНФ:

$$\overline{x_1} \cdot x_2 \vee x_1 \cdot \overline{x_2} \vee x_1 \cdot x_2 = Y.$$

Розташуємо логічні одиниці у всіх квадратах, яким відповідають добутки у вихідній формулі на рисунку 1.3. Заповнена таким чином карта Карно тепер готова для побудови. Сусідні одиниці об'єднуються в один контур групами по дві, чотири або вісім одиниць. Побудова контурів продовжується до тих пір, поки всі одиниці не опиняться всередині контурів. Кожний контур - це новий член спрощеної функції перемикання. Відмітимо, що на рисунку 4 получилось тільки два контури. Це означає, що нова, спрощена функція перемикання буде складатися тільки з двох членів, що пов'язані функцією АБО.

	$\overline{x_2}$	x_2
$\overline{x_1}$		1
x_1	1	1

Рисунок 1.3 - Нанесення одиниць на карту Карно

	$\overline{x_2}$	x_2
$\overline{x_1}$		1
x_1	1	1

Рисунок 1.4 - Об'єднання одиниць групами в один контур

Тепер спростимо функцію, приймаючи до уваги два контури на рисунку 1.4. Взявши спочатку нижній контур, замітимо, що x_1 тут зустрічається у комбінації з x_2 і $\overline{x_2}$. У відповідності з правилами алгебри логіки x_2 і $\overline{x_2}$ доповнюють один одного і їх можна опустити. Тоді в нижньому контурі залишається тільки член x_1 . Аналогічно цьому вертикально розташований контур вміщує x_1 і $\overline{x_1}$, які також можна опустити, залишивши тільки x_2 . Елементи x_1 і x_2 , що залишилися, об'єднуються функцією АБО, що приводить до спрощеної функції перемикання $x_1 \vee x_2 = Y$. Алгоритм мінімізації функції записується таким чином:

1. Переведення функції в ДДНФ.
2. Нанесення одиниць на карту Карно.
3. Об'єднання сусідніх одиниць контурами, що охоплюють два, чотири або вісім квадратів.
4. Проведення спрощення, виключаючи члени, які доповнюють один одного всередині

контур.

5. Об'єднання членів, що залишилися (по одному у кожному контурі), функцією АБО.

6. Запис мінімізованої функції в ДДНФ.

Карта Карно з трьома змінними. Розглянемо функцію

$$x_1 \cdot \overline{x_2} \cdot \overline{x_3} \vee \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \vee \overline{x_1} \cdot \overline{x_2} \cdot x_3 \vee x_1 \cdot \overline{x_2} \cdot x_3 = Y.$$

Карта Карно для цієї функції показана на рисунку 5. Нижній контур вміщує x_2 і $\overline{x_2}$, їх можна опустити. Після цього у складі нижнього контуру залишається тільки член $\overline{x_1} \cdot \overline{x_3}$. У верхній контур входять x_3 і $\overline{x_3}$, які теж опускаються, після чого залишається тільки член $\overline{x_1} \cdot \overline{x_2}$. Підсумкова функція має вигляд $\overline{x_1} \cdot \overline{x_3} \vee \overline{x_1} \cdot \overline{x_2} = Y$.

Суттєво, щоб карта Карно була складена саме так, як показано на рисунку 1.5. Замітимо, що в міру того, як зміщатись вниз по лівій частині карти, на кожному кроці змінюється лише одна змінна. Зверху зліва записаний добуток $\overline{x_1} \cdot \overline{x_2}$, а рядком нижче $\overline{x_1} \cdot x_2$ (заміна $\overline{x_2}$ на x_2). Далі, при пересуванні від $\overline{x_1} \cdot \overline{x_2}$ до $\overline{x_1} \cdot x_2$ вниз $\overline{x_1}$ переходить в x_1 і т.д. Якщо карту Карно скласти неправильно, то вона не принесе очікуваного результату.

	$\overline{x_3}$	x_3
$\overline{x_1} \cdot \overline{x_2}$	1	1
$\overline{x_1} \cdot x_2$		
$x_1 \cdot x_2$	1	
$x_1 \cdot \overline{x_2}$	1	

	$\overline{x_3}$	x_3
$\overline{x_1} \cdot \overline{x_2}$	1	1
$\overline{x_1} \cdot x_2$		
$x_1 \cdot x_2$	1	
$x_1 \cdot \overline{x_2}$	1	

Рисунок 1.5 - Спрощення перемикаючої функції на основі карти Карно

Тема №1 Логіко-математичні основи інформатики

Практичне заняття: Системи числення

Мета заняття: Ознайомитись з системами числення (2, 8, 16)

Час проведення: 2 год

Навчальні питання:

1. Системи числення
2. Шістнадцяткові числа

Системи числення

Цифрові обчислювальні машини працюють із двійковими числами. Двійкова система числення або система з основою 2 використовує тільки цифри 0 і 1. Ці двійкові числа названі бітами (від *binary digit*). Фізично в цифрових електронних системах біт 0 представлений напругою LOW (низьким), а біт 1 — напругою HIGH (високим) ¹.

Людська діяльність припускає використання десяткової системи числення. Десяткова система, або система з основою 10, містить 10 цифр (від 0 до 9). Вона також характеризується значенням позиції (або вагою). У табл. 1 показано, наприклад, що десяткове число 1327 дорівнює одній тисячі, плюс три сотні, плюс два

десятки, плюс сім одиниць ($1000+300+20+7=1327$).

Двійкова система володіє також властивістю зрівноважування. У табл. 2 наведені десяткові значення перших чотирьох двійкових позицій. Двійкове число 1001 (вимовляється: один, нуль, нуль, один) перетворено, таким чином, у свій десятковий еквівалент 9. Біт одиниці двійкового числа в табл. 2 називається молодшим бітом (МБ), біт вісімки - старшим бітом (СБ).

Таблиця 1. Значення позицій десяткових чисел

Ступінь основи	10^3	10^2	10^1	10^0
Значення позицій	1000	100	10	1
Десяткові	$\left\{ \begin{array}{l} 1 \quad 3 \quad 2 \quad 7 \\ 1000 + 300 + 20 + 7 = 1327 \end{array} \right.$			

Таблиця 2. Значення позицій двійкових чисел

Ступінь основи	2^3	2^2	2^1	2^0
Значення позицій	8	4	2	1
	СБ			МБ
Двійкове	1	0	0	1
Десяткове	$8 + 0 + 0 + 1 = 9$			

У табл. 3 наведені десяткові числа від 0 до 15, а також їхні двійкові еквіваленти. Ті, хто працює в області використання ЕОМ, повинні, щонайменше, запам'ятати ці двійкові числа.

Таблиця 3. Десяткові числа і їхні двійкові еквіваленти

Десяткові	Двійкові				Десяткові	Двійкові			
10 1	8	4	2	1	10 1	8	4	2	1
0				0		8			
1				1		9			
2			1	0	1	0			
3			1	1	1	1			
4		1	0	0	1	2			
5		1	0	1	1	3			
6		1	1	0	1	4			
7		1	1	1	1	5			

Як перетворити двійкове число 1011 0110 (тобто один, нуль, один, один, нуль, один, один, нуль) у його десятковий еквівалент? Процедура перетворення виконується відповідно до табл. 4. Десяткові значення кожної позиції записані під кожним бітом, потім десяткові числа підсумуються ($128 + 32 + 16 + 4 + 2 = 182$), що дає 182.

Таблиця 4. Двійково-десяткові перетворення

Ступінь основи	2^7	2^6	2^5		2^4	2^3	2^2		2^1	2^0	
Значення позицій	128	64	32		16	8	4		2	1	
Двійкове	1	0	1		1	0	1		1	0	
Десяткове	128	+	32	+	16	+	4	+	2	=	182_{10}

Звичайна основа системи числення вказується індексами. Таким чином, число $1011\ 0110_2$ є двійковим (або основи 2), а число 182_{10} — десятковим: $1011\ 0110_2 = 182_{10}$.

Як перетворити десяткове 155 у його двійковий еквівалент? Процедура перетворення наведена на мал.

$155 : 2 = 77$ залишок 1^{MB}
 $77 : 2 = 38$ залишок 1
 $38 : 2 = 19$ залишок 0
 $19 : 2 = 9$ залишок 1
 $9 : 2 = 4$ залишок 1
 $4 : 2 = 2$ залишок 0
 $2 : 2 = 1$ залишок 0
 $1 : 2 = 0$ залишок 1^{CB}

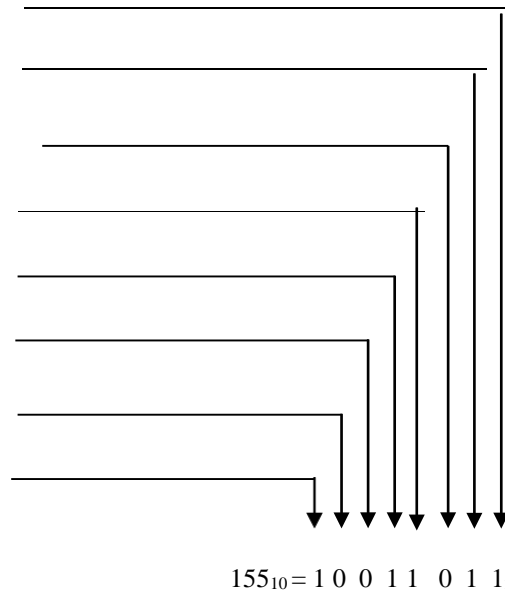


Рис 1. Двійково-десятькове перетворення

Десяткове 155 спочатку ділиться на 2, що дає нам частку 77 і залишок 1. Цей залишок стає МБ двійкового числа й міститься в цю позицію (див. мал. 1). Потім частка (77) переміщається і стає наступним діленням. Потім кожна частка послідовно ділиться на 2 доти, поки не вийде частка, рівна 0, і залишок, рівний 1 (див. передостанній рядок на мал. 1). Останній рядок на мал. 1. дає нам результат $155_{10} = 10011011_2$.

Шістнадцяткові числа

Комірка пам'яті типової мікро-ЕОМ може містити двійкове число 1001 1110. Такий довгий ланцюг нуліві одиниць складний для запам'ятовування й незручний для введення із клавіатури. Число 1001 1110 могло б бути перетворене в десяткове, що дало б 158_{10} , але процес перетворень зайняв би багато часу. Більша частина систем мікроінформатики використовує шістнадцяткову форму запису, щоб спростити запам'ятовування й використання таких двійкових чисел, як 1001 1110.

Шістнадцяткова система числення (hexadecimal), або система з основою 16, використовує 16 символів від 0 до 9 і A, B, C, D, E, F. У табл. 5 наведені еквіваленти десяткових, двійкових і шістнадцяткових чисел. Помітимо з табл. 5, що кожний шістнадцятковий символ може бути представлений єдиним сполученням чотирьох біт. Таким чином, поданням двійкового числа 1001 1110 у шістнадцятковому коді є число 9E. Це значить, що частина 1001 двійкового числа дорівнює 9, а частина 1110 дорівнює E (звичайно, у шістнадцятковому коді). Отже, $10011110_2 = 9E_{16}$. (Не варто забувати що індекси означають основу системи числення.)

Як перетворити двійкове число 111010 у шістнадцяткове? Треба почати із МБ і розділити двійкове числі на групи з 4 біт. Потім треба замінити кожен групу з 4 біт еквівалентною шістнадцятковою цифрою $1010_2 = A$, $0011_2 = 3$, отже, $111010_2 = 3A_{16}$.

Як перетворити шістнадцяткове число 7F у двійкове? У цьому випадку кожна шістнадцяткова цифра повинна бути замінена своїм двійковим еквівалентом з 4 біт. У прикладі двійкове число 0111 замінено шістнадцятковою цифрою 7, а 1111_2 замінює F_{16} , звідки $7F_{16} = 01111111_2$.

Шістнадцятковий запис широко використовується для подання двійкових чисел, тому необхідно табл. 5 також запам'ятати.

Таблиця 5. Десяткові, шістнадцяткові й двійкові еквіваленти

Десяткові	Шістнадцяткові	Двійкові			
		8	4	2	1
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1
10	A	1	0	1	0
11	B	1	0	1	1
12	C	1	1	0	0
13	D	1	1	0	1
14	E	1	1	1	0
15	F	1	1	1	1

Таблиця 6. Перетворення шістнадцяткового числа в десяткове

Ступінь шістнадцяти	16^3		16^2		16^1		16^0	
Значення позицій	4096		256		16		1	
Шістнадцяткове	2		C		6		E ₁₆	
Десяткове	4096 × 2		256 × 12		16 × 6		1 × 14	=11374 ₁₀
	8192	+	3072	+	96	+	14	

Перетворимо шістнадцяткове число 2C6E в десяткове. Процедура дій відповідає табл. 6. Значеннями позицій перших чотирьох шістнадцяткових цифр є відповідно ліворуч праворуч 4096, 256, 16 і 1. Десяткове число містить 14 (E₁₆) одиниць, 6 чисел 16, 12 (C₁₆) чисел 256 і 2 числа 4096. Кожна цифра множиться на відповідну ййвагу, виходить сума, що і дає нам десяткове число 11374.

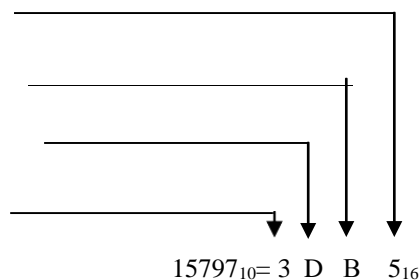
Перетворимо десяткове число 15797 у шістнадцяткове. На мал. 2. показана процедура дій.

$15797_{10} : 16 = 987$ залишок $5_{10} = 5_{16}^{MP}$

$987_{10} : 16 = 61$ залишок $11_{10} = B_{16}$

$61_{10} : 16 = 3$ залишок $13_{10} = D_{16}$

$3_{10} : 16 = 0$ залишок $3_{10} = 3_{16}^{CP}$

**Рис. 2** Десятково-шістнадцяткове перетворення.

У першому рядку 15797_{10} поділено на 16, що дає частку 987_{10} залишок 5_{10} , що перетвориться потім у свій шістнадцятковий еквівалент ($5_{10} = 5_{16}$) і стає цифрою молодшого розряду (MP) шістнадцяткового числа. Перша частка (987) стає діленням у другому рядку й знову ділиться на 16, що дає частку 61 і залишок 11_{10} або шістнадцяткове B. У третьому рядку 61 ділиться на 16 дає частка 3 і залишок 13_{10} або D₁₆, а в четвертій рядків ділене 3 ділиться на 16, дає частка 0 і залишок 3_{10} або 3₁₆. Коли частка дорівнює 0, як у четвертому рядку, перетворення закінчується. 3₁₆ стає цифрою старшого розряду (CP) результату, тобто 3DB5₁₆.

Література [3,с.20-45], [13, с.37-89], [15, с.45-80]

Тема№1 Логіко-математичні основи інформатики

Практичне заняття: Контрольна робота

Мета заняття: Здійснити контроль знань за темою «Логіко-математичні основи інформатики»

Час проведення: 2 год

Тема№4 Програмне забезпечення

Практичне заняття: Операційна система Windows. Реєстр Windows

Мета заняття: Ознайомитися з роботою операційних ситем

Час проведення: 2 год

Навчальні питання:

1. Програмне забезпечення
2. Операційні системи
3. Управління даними
4. Реєстр операційної системи Windows.

1. Програмне забезпечення

В основу роботи комп'ютерів покладено програмний принцип управління, який полягає в тому, що комп'ютер виконує дії за заздалегідь заданою програмою. Цей принцип забезпечує універсальність використання комп'ютера: у певний момент часу розв'язується задача відповідно до вибраної програми. Після її завершення в пам'ять завантажуються інша програма і т.д.

Програма - це запис алгоритму розв'язування задачі у вигляді послідовності команд або операторів мовою, яку розуміє комп'ютер. Кінцевою метою будь-якої комп'ютерної програми є управління апаратними засобами.

Для нормального розв'язування задач на комп'ютері потрібно, аби програма була налагоджена, не вимагала доопрацювань і мала відповідну документацію. Тому, щодо роботи на комп'ютері часто використовують термін

програмне забезпечення (software), під яким розуміється сукупність програм, процедур, правил, а також документації, що стосуються функціонування системи обробки даних.

Програмне та апаратне забезпечення в комп'ютері працюють у нерозривному зв'язку і взаємодії. Склад програмного забезпечення обчислювальної системи називається програмною конфігурацією. Між програмами існує взаємозв'язок, тобто робота множини програм базується на програмах нижчого рівня.

Міжпрограмний інтерфейс - це розподіл програмного забезпечення на декілька пов'язаних між собою рівнів. Рівні програмного забезпечення є пірамідою, де кожний вищий рівень базується на програмному забезпеченні попередніх рівнів. Схематично структура програмного забезпечення наведена на рис. 4.1.

прикладний рівень
службовий рівень
системний рівень
базовий рівень

Рис. 4.1 - Рівні програмного забезпечення

Базовий рівень є нижчим рівнем програмного забезпечення. Він відповідає за взаємодію з базовими апаратними засобами. Базове програмне забезпечення міститься у складі базового апаратного забезпечення і зберігається у спеціальних мікросхемах постійної пам'яті (ПЗП), утворюючи базову систему введення-виведення BIOS. Програми та дані записуються в ПЗП на етапі виробництва і не можуть бути змінені під час експлуатації.

Системний рівень - є перехідним. Програми цього рівня забезпечують взаємодію інших програм комп'ютера з програмами базового рівня і безпосередньо з апаратним забезпеченням. Від програм цього рівня залежать експлуатаційні показники всієї обчислювальної системи. При під'єднанні до комп'ютера нового обладнання на системному рівні повинна бути встановлена програма, яка забезпечує для решти програм взаємозв'язок із пристроєм. Конкретні програми, призначені для взаємодії з конкретними пристроями, називаються драйверами.

Інший клас програм системного рівня відповідає за взаємодію з користувачем. Завдяки йому, можна вводити дані в обчислювальну систему, управляти її роботою й одержувати результат у зручній формі. Це засоби забезпечення користувацького інтерфейсу, від них залежить зручність та продуктивність роботи з комп'ютером.

Сукупність програмного забезпечення системного рівня утворює ядро операційної системи комп'ютера. Наявність ядра операційної системи - це перша умова для можливості практичної роботи користувача з обчислювальною системою. Ядро операційної системи виконує такі функції: управління пам'яттю, процесами введення-виведення, файловою системою, організацією взаємодії та диспетчеризації процесів, облік використання ресурсів, опрацювання команд тощо.

Програми службового рівня взаємодіють як із програмами базового рівня, так і з програмами системного рівня. Призначення службових програм (утиліт) полягає в автоматизації робіт із перевірки та налаштування комп'ютерної системи, а також для покращення функцій системних програм. Деякі службові програми (програми обслуговування) відразу входять до складу операційної системи, доповнюючи її ядро, але більшість єзовнішніми програмами і розширюють функції операційної системи. Тобто, в розробці службових програм відслідковуються два напрямки: інтеграція з операційною системою та автономне функціонування.

2. Операційні системи

Операційна система (ОС, operating system, OS) – це комплекс управляючих і опрацьовуючих програм, які, з одного боку, виступають як інтерфейс між пристроями обчислювальної системи і прикладними програмами, а з іншого боку - призначені для управління пристроями, управління обчислювальними процесами, управління ефективним розподілом обчислювальних ресурсів між обчислювальними процесами і організації надійних обчислень. Це визначення застосовне до більшості сучасних операційних систем загального призначення.

Є застосування обчислювальної техніки, для яких операційні системи не потрібні. Наприклад, вбудовані мікрокомп'ютери, що містяться у багатьох побутових приладах, автомобілях (іноді до десятка в кожному), найпростіших стільникових телефонах, постійно виконують лише одну програму, яка завантажується при ввімкненні. Чимало простих ігрових приставок також є спеціалізованими мікрокомп'ютерами і можуть обходитися без операційної системи, завантажуючи при ввімкненні програму, записану на вставленому в пристрій «картриджі» або компакт-диску.

Операційні системи потрібні, якщо:

- обчислювальна система використовується для різних задач, причому програми, які розв'язують ці задачі, потребують збереження даних та обміну ними. З цього випливає необхідність універсального механізму збереження даних; у переважній більшості випадків в операційних систем ця необхідність реалізується файловою системою. Сучасні операційні системи, крім того, надають можливість безпосередньо «зв'язати» результат однієї програми із введенням іншої, минаючи відносно повільні дискові операції;

- різні програми потребують виконання одних і тих же рутинних дій. Наприклад, просте введення символу з клавіатури і відображення його на екрані може вимагати виконання сотень машинних команд, а дискова операція - тисяч. Аби не програмувати їх щоразу заново, операційні системи надають системні бібліотеки часто використовуваних підпрограм (функцій);

- між програмами та користувачами системи необхідно розподіляти повноваження, аби користувачі могли захищати свої дані від несанкціонованого доступу, а можлива помилка в програмі не викликала тотальних неприємностей;

- необхідна можливість імітації «одночасного» виконання декількох програм на одному комп'ютері (якщо він містить лише один процесор), що здійснюється за допомогою прийому, відомого як «розподіл часу». При

цьому спеціальний компонент, який називається планувальником, ділить процесорний час на короткі відрізки і надає їх по черзі різним виконуваним програмам (процесам);

- оператор повинен мати можливість так чи інакше управляти процесами виконання окремих програм. Для цього використовуються операційні середовища - оболонки і набори утиліт - вони можуть бути частиною операційної системи.

Таким чином, сучасні універсальні операційні системи можна охарактеризувати, насамперед, як такі, що:

- використовують файлові системи (з універсальним механізмом доступу до даних),
- є багатокористувацькими (із розподілом повноважень),
- є багатозадачними (із розподілом часу).

У логічній структурі типової обчислювальної системи операційна система займає положення між пристроями з їх мікроархітектурою, машинною мовою, можливо, власними (вбудованими) мікропрограмами - з одного боку - і прикладними програмами з іншого.

Розробникам програмного забезпечення операційна система дозволяє абстрагуватися від деталей реалізації та функціонування пристроїв, надаючи мінімально необхідний набір функцій.

У більшості обчислювальних систем операційна система є основною, найбільш важливою (а іноді і єдиною) частиною системного програмного забезпечення. З 1990-х років найбільш поширеними операційними системами є системи ряду Windows і системи класу UNIX (особливо Linux і Mac OS).

При розгляді засад функціонування ОС прийнято виділяти чотири основні групи функцій, виконуваних системою:

- управління пристроями; маються на увазі всі периферійні пристрої, що вмикаються до комп'ютера, - клавіатура, монітор, принтери, диски і т.п.;
- управління даними; під цим терміном розуміється робота з файлами;
- управління процесами; ця сторона роботи ОС пов'язана із завантаженням і завершенням роботи програм, обробкою помилок, забезпеченням паралельної роботи декількох програм на одному комп'ютері;
- управління пам'яттю; оперативна пам'ять комп'ютера - це такий ресурс, якого завжди не вистачає; у цих умовах розумне планування використання пам'яті є найважливішим чинником ефективної роботи.

Є ще кілька важливих обов'язків, які лягають на ОС і які важко втиснути в рамки традиційної класифікації функцій. До них, насамперед, відносяться такі:

- організація інтерфейсу з користувачем; форми інтерфейсу можуть бути різноманітними, залежно від типу і призначення ОС: мова управління пакетами завдань, набір діалогових команд, засоби графічного інтерфейсу;
- захист даних; як тільки система перестає бути надбанням одного ізольованого від зовнішнього світу користувача, питання захисту даних від несанкціонованого доступу набувають першорядну важливість; ОС, яка забезпечує роботу в мережі або в системі розподілу часу, повинна відповідати наявним стандартам безпеки;
- ведення статистики; у ході роботи ОС повинна збиратися, зберігатися і аналізуватися різноманітна інформація: про кількість часу, витраченого різними програмами та користувачами, про інтенсивність використання ресурсів, про спроби некоректних дій користувачів, про збої устаткування і т.п.; зібрана інформація зберігається в системних журналах і в облікових записах користувачів.

Для розуміння роботи ОС необхідно вміти виділяти основні частини системи та їх зв'язки, тобто описувати структуру системи. Для різних ОС їх структурний поділ може бути досить різним. Найбільш загальними видами структуризації можна вважати два. З одного боку, можна вважати, що ОС розділена на підсистеми, відповідні перерахованим вище групам функцій. Такий поділ досить обґрунтований, програмні модулі ОС дійсно в основному можна віднести до однієї з цих підсистем. Інший важливий структурний поділ пов'язаний із поняттям ядра системи.

Ядро, як можна зрозуміти з назви, це основна, «найсистемніша» частина операційної системи. Є різні визначення ядра. Відповідно до одного з них, ядро - це резидентна частина системи, тобто до ядра належить той програмний код, який постійно знаходиться в пам'яті протягом всієї роботи системи. Інші модулі ОС є транзитними, тобто вони довантажуються в пам'ять із диска в міру необхідності на час своєї роботи. До транзитних частин системи відносяться:

- утиліти (utilities) - окремі системні програми, які вирішують окремі завдання, такі як форматування і перевірку дисків, пошук даних у файлах, моніторинг (відстеження) роботи системи та багато іншого;
- системні бібліотеки підпрограм, які дозволяють прикладним програмам використовувати різні спеціальні можливості, підтримувані системою (наприклад, бібліотеки для графічного виведення, для роботи з мультимедіа тощо);
- інтерпретатор команд - програма, що виконує введення команд користувача, їх аналіз і виклик з інших модулів для виконання команд;
- системний завантажувач - програма, яка при завантаженні ОС (наприклад, при ввімкненні живлення) забезпечує завантаження системи з диска, її ініціалізацію і старт;
- інші види програм у залежності від конкретної системи.

Як основоположний елемент операційної системи, ядро є найбільш низьким рівнем абстракції для доступу програм до ресурсів обчислювальної системи, необхідних для їх роботи. Як правило, ядро надає такий доступ виконуваним процесам відповідних застосувань за рахунок використання механізмів взаємодії між процесами та звертанням застосувань до системних викликів ОС.

Описана задача може різнитися залежно від типу архітектури ядра і способу її реалізації.

3. Управління даними

Файлова система - це частина операційної системи, призначення якої полягає в тому, аби забезпечити

користувачеві зручний інтерфейс при роботі з даними, що зберігаються на диску, і забезпечити спільне

використання файлів кількома користувачами і процесами.

У широкому розумінні поняття "файлова система" включає:

- сукупність усіх файлів на диску;
- набори структур даних, що використовуються для управління файлами, такі, наприклад, як каталоги файлів, дескриптори файлів, таблиці розподілу вільного і зайнятого простору на диску;
- комплекс системних програмних засобів, які реалізують управління файлами, зокрема, створення, знищення, читання, запис, іменування, пошук та інші операції над файлами.

Файл – це з одного боку, формально не визначене поняття. З точки зору змісту, файл – це множина даних, об'єднаних деяким логічним зв'язком, тобто одна і та ж сукупність даних може розглядатися як один або декілька файлів (початкові дані до задачі; навчальні матеріали тощо). З іншого боку, поняття файл відноситься до об'єкту, який однозначно визначається такими ознаками:

- файл об'єднує множину даних;
- має ім'я;
- з метою довготривалого та надійного зберігання інформації розташовується на зовнішньому пристрої;
- передбачає багаторазове використання інформації з розривом у часі;
- передбачає спільне використання інформації декількома застосуваннями або користувачами одночасно (розподіляється ресурс) або з розривом у часі.

Прийняте означення є означенням файлу з точки зору користувача. Деталі розташування даних на зовнішньому пристрої і роботи з ними на низькому (фізичному) рівні файлова система бере на себе, екрануючи всі складнощі цього рівня і надаючи користувачеві зручну логічну модель і набір відповідних команд.

Користувачі дають файлам символічні імена, при цьому враховуються обмеження операційної системи як на використовувані символи, так і на довжину імені.

Файли бувають різних типів: звичайні файли, спеціальні файли, файли-каталоги.

Звичайні файли, у свою чергу, діляться на текстові та виконувані. Текстові файли складаються з рядків символів, поданих у ASCII-коді. Це -документи, початкові тексти програм тощо. Текстові файли можна прочитати на екрані і надрукувати на принтері. Двійкові файли не використовують ASCII-коди, вони часто мають складну внутрішню структуру. Всі операційні системи повинні вміти розпізнавати принаймні один тип файлів - їх власні виконувані файли.

Спеціальні файли - це файли, асоційовані з пристроями введення-виведення, які дозволяють користувачеві виконувати операції введення-виведення, використовуючи звичайні команди запису у файл або читання з файлу. Ці команди обробляються спочатку програмами файлової системи, а потім на деякому етапі виконання запиту перетворюються операційною системою в команди управління відповідним пристроєм. Спеціальні файли, так само як і пристрої введення-виведення, діляться на блок-орієнтовані і байт-орієнтовані.

Каталог - це, з одного боку, група файлів, об'єднаних користувачем виходячи з деяких міркувань, а з іншого боку - це файл, що містить системну інформацію про групу файлів, його складових. У каталозі міститься список файлів, що входять до нього, і встановлюється відповідність між файлами і їх характеристиками (атрибути).

Реєстр операційної системи Windows.

Системний реєстр (реєстр операційної системи Windows) - це база даних, яку використовують операційні системи ряду Windows для зберігання відомостей про конфігурацію комп'ютера.

Реєстр містить відомості, до яких операційна система ряду Windows постійно звертається під час роботи, а саме:

- профілі всіх користувачів;
- дані про встановлені програми і типи документів, створюваних кожною програмою;
- значення властивостей для папок і значків програм;
- конфігурація устаткування, встановленого в операційній системі;
- дані про використовувані порти.

Реєстр має ієрархічну деревоподібну структуру, що складається з розділів (табл. 4.1), підрозділів і ключів.

Таблиця 4.1.

Ім'я кореневого розділу	Опис
HKEY_LOCAL_MACHINE	Відомості про локальний комп'ютер, включаючи дані про обладнання і операційну систему, такі як тип шини, системна пам'ять, драйвери пристроїв і параметри завантаження.
HKEY_CLASSES_ROOT	Відомості, що використовуються різними технологіями OLE, і дані про зіставлення типів файлів. Певний розділ або параметр існує в HKEY_CLASSES_ROOT, якщо відповідний розділ або параметр існує в HKEY_LOCAL_MACHINE \ SOFTWARE\Classes або HKEY_CURRENT_USER\SOFTWARE\Classes. Якщо розділ або
	параметр є в обох місцях, в HKEY_CLASSES_ROOT з'явиться значення з HKEY_CURRENT_USER.

HKEY_CURRENT_USER	Профіль користувача, що увійшов у систему локально (на відміну від віддаленого користувача), включаючи змінні середовища, параметри робочого столу, мережевих підключень, принтерів і застосувань. Це піддерево є псевдонімом піддерева HKEY_USERS і вказує на HKEY_USERS\обліковий_код_поточного_користувача.
HKEY_USERS	Відомості про завантажені профілі користувача і профіль, використовуваний за умовчуванням. Сюди включені відомості, що з'являються в піддереві HKEY_CURRENT_USER. Віддалені користувачі не мають профілів у цьому розділі серверу; їхні профілі знаходяться у реєстрах власних комп'ютерів.
HKEY_CURRENT_CONFIG	Відомості про профіль обладнання, яке використовується локальним комп'ютером при запуску системи. Ці відомості використовуються для налаштування завантажувальних драйверів і спроможності дисплея. Це піддерево є частиною піддерева HKEY_LOCAL_MACHINE і відповідає HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current.

У деяких випадках відновити працездатність системи після збою можна, завантаживши працездатну версію реєстру. Для цього необхідно мати копію реєстру. Основним засобом для переглядання і редагування записів реєстру є спеціальна утиліта "Редактор реєстру".

Файл редактора реєстру знаходиться в папці Windows і називається regedit.exe.

У деяких випадках для лікування зараженого вірусом комп'ютера і\або усунення наслідків зараження комп'ютера вірусом необхідно редагувати\вносити зміни до реєстру операційної системи ряду Windows, яка встановлена на зараженому комп'ютері.

Аби відкрити системний реєстр необхідно виконати такі дії:

- натискується кнопка Пуск;
- вибирається меню Виконати;
- у полі вводиться REGEDIT;
- натискується кнопка ОК.

Робота з розділами реєстру аналогічна роботі з папками у Провіднику Windows.

У реєстрі Windows системна інформація розбита на так звані кущі (hive). Термін «кущ» описує деревоподібну структуру розділів, підрозділів і параметрів, що виходить із вершини ієрархії реєстру. Куш міститься в окремому файлі і має окремий журнал, що знаходяться у папках Windows\System32\Config або C:\Documents and Settings\ім'я користувача.

Кожний куш у реєстрі Windows пов'язаний із набором стандартних файлів. Імена стандартних кущів і файлів показані в табл. 4.2.

Таблиця 4.2.

Куш реєстру	Імена файлів
HKEY_LOCAL_MACHINE \	SAM Sam і Sam.log
HKEY_LOCAL_MACHINE \	SECURITY Security і Security.log
HKEY_LOCAL_MACHINE \	SOFTWARE Software і Software.log
HKEY_LOCAL_MACHINE \ SYSTEM	System і System.log
HKEY_CURRENT_CONFIG	System і System.log
HKEY_CURRENT_USER	Ntuser.dat і Ntuser.dat.log
HKEY_USERS \ .DEFAULT	Default і Default.log

Список імен кущів і шляхів до каталогів, в яких вони зберігаються, розташовані в розділі HKEY_LOCAL_MACHINE\CurrentControlSet\Control\hivelist (рис. 4.17).

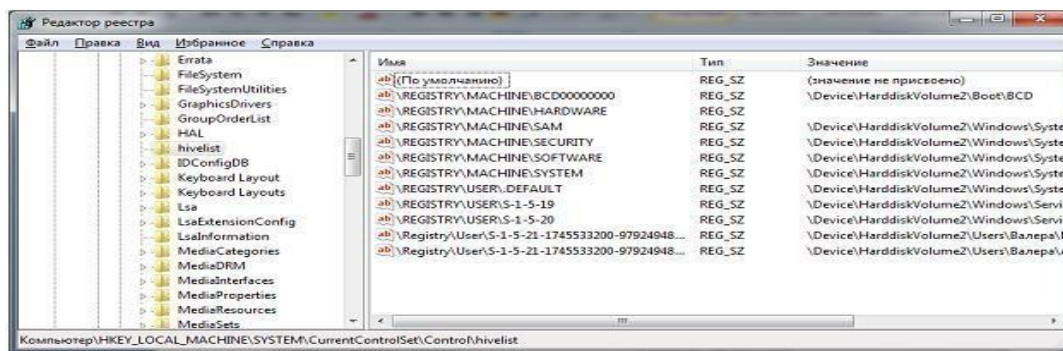


Рис. 4.17.

Кожний розділ або підрозділ може містити дані, які називаються параметрами (ключами). Деякі параметри зберігають відомості для конкретних користувачів, інші зберігають відомості, що стосуються всіх користувачів комп'ютера. Параметр реєстру має ім'я, тип даних і значення.

Наступна таблиця (табл. 4.3) містить список даних, визначених і використовуваних системою.

Реєстр містить важливі дані про комп'ютер, його застосування і файли. Зловмисник може скористатися реєстром для нанесення серйозного збитку комп'ютеру. Досить важливо підтримувати високий рівень безпеки реєстру.

Таблиця 4.3.

Тип даних	Опис
REG_BINARY	Неопрацьовані двійкові дані. Більшість відомостей про устаткування зберігається у вигляді двійкових даних і виводиться в редакторі реєстру в шістнадцятиричному форматі.
REG_DWORD	Дані, які подаються цілим числом (4 байти). Багато параметрів служб і драйверів пристроїв мають цей тип і обробляються в двійковому, шістнадцятиричному або десятковому форматах.
REG_EXPAND_SZ	Рядок даних змінної довжини. Цей тип даних включає змінні, оброблювані при використанні програми або служби.
REG_MULTI_SZ	Складний рядок. Цей тип, як правило, має списки та інші записи у форматі, зручному для читання. Записи відокремлюються комами, прогалинами або іншими символами.
REG_SZ	Текстовий рядок фіксованої довжини.
REG_FULL_RESOURCE_DESCRIPTOR	Низка вкладених масивів, розроблена для зберігання списку ресурсів апаратного компонента або драйвера.

Типово адміністраторам надається повний доступ до всього реєстру, в той час як інші користувачі, в основному, мають повний доступ до розділів, що належать до їх власних облікових записів (у тому числі HKEY_CURRENT_USER) і доступ на читання до розділів, які належать до комп'ютера і його програмного забезпечення. Користувачі не мають доступу до розділів, що відносяться до облікових записів інших користувачів. Користувачі, що мають відповідні дозволи доступу до розділів, можуть змінювати дозволи на доступ до цього розділу і до будь-яких інших, що містяться у цьому розділі.

Важливо підкреслити, що **одна неправильна зміна в реєстрі може вивести операційну систему з ладу**.

При редагуванні/зміні реєстру варто скористатися такими порадами:

- перед внесенням змін до реєстру потрібно робити резервну копію розділу або підрозділу, в який будуть вноситися зміни;
- не можна реєстр Windows XP замінити реєстром іншої версії операційних систем Windows;
- ніколи не залишайте редактор реєстру працюючим без нагляду.

. Облікові записи користувачів в ОС Windows

Обліковий запис – це сукупність даних про користувача, яка зберігається в комп'ютерній системі, і яка необхідна для розпізнавання (аутентифікації) користувача і надання доступу до його особистих даних і налаштувань. За синонім також використовуються акк, аккаунт і екаунт.

Для використання облікового запису (іншими словами, для входу в систему під чийсь іменем), як правило, потрібне введення імені (логіну) і пароля. Також може вимагатися інша додаткова інформація.

Користувачі мережі Internet можуть сприймати обліковий запис як особисту сторінку, профіль, кабінет,

місце зберігання особистих та інших відомостей на певному інтернет-ресурсі.

Обліковий запис, як правило, містить відомості, необхідні для впізнання користувача при підключенні до системи, відомості для авторизації і обліку. Це ідентифікатор користувача (login) і його пароль. Пароль або його аналог, як правило, зберігається в зашифрованому або хешірованому вигляді для забезпечення його безпеки.

Для аутентифікації можуть також використовуватися спеціальні файли-ключі (які можна зберігати на різних носіях інформації) або апаратні засоби (виробляють одноразові ключі, зчитувальні біометричні характеристики тощо), а також одноразові паролі.

Для підвищення надійності, поряд із ключем або паролем можуть бути передбачені інші засоби зв'язки, наприклад, особливі потайні запитання (або декілька запитань) такого змісту, на які відповідь може бути відомою тільки користувачеві. Такі запитання і відповіді також зберігаються в обліковому записі.

Обліковий запис може містити додаткові опитувальні дані про користувача - ім'я, прізвище, по батькові, псевдонім, стать, народність, расову приналежність, віросповідання, групу крові, резус-фактор, вік, дату народження, адресу e-mail, домашню адресу, робочу адресу, номер домашнього телефону, номер робочого телефону, номер стільникового телефону, номер ICQ (ICQ - централізована служба миттєвого обміну повідомленнями мережі Internet), ідентифікатор Skype, «нік» в IRC (IRC - протокол прикладного рівня для обміну повідомленнями в режимі реального часу), адресу домашньої сторінки та/або блогу в Павутині чи інtranеті, відомості про захоплення, про коло інтересів, про сім'ю, про перенесені хвороби, про політичні уподобання, про партійну приналежність, про культурні уподобання, про вміння спілкуватися іноземними мовами тощо. Конкретні категорії даних, які можуть бути внесені в такий опитувальник, визначаються авторами і (або) адміністраторами системи.

Обліковий запис може також містити одну або декілька фотографій чи аватар користувача. Аватар - графічне подання користувача, його alter ego або ігрового персонажа. Аватар може бути двовимірним зображенням (іконкою) у веб-форумах, чатах, порталах, блогах, соціальних мережах, або ж бути тривимірною моделлю (віртуальні світи, масові багатокористувацькі онлайн ігри). Крім того, аватар може бути поданий у вигляді тексту. Основна мета аватара - публічне графічне подання користувача, створене самим користувачем.

Обліковий запис користувача також може враховувати різні статистичні властивості поведінки користувача в системі: давність останнього входу в систему, тривалість останнього перебування в системі, адресу використаного комп'ютера, частотність використання системи, загальна і (або) питома кількість певних дій, вироблених в системі тощо.

Існують три типи облікових записів. Кожний тип дає користувачеві різні можливості для управління комп'ютером:

- стандартні облікові записи користувачів призначені для повсякденної роботи;
- облікові записи адміністратора надають повний контроль над комп'ютером і застосовуються тільки в необхідних випадках;
- облікові записи гостя призначені для тимчасового доступу до комп'ютера.

Інформаційне та методичне забезпечення:

Основна література

1. Основи математичної логіки: навчальний посібник/Зубенко В.В., Шкільняк С.С.. К.: НУБіП України, 2020. - 102 с.
2. Математична логіка та теорія алгоритмів: Лекції: навч. посіб. для студ. спеціальності 124 «Системний аналіз» / О. В. Стусь ; КПІ ім. Ігоря Сікорського. 2017. – 150 с.
3. Дискретна математика : навч. посіб. / уклад. : С. І. Балага; рец. : О. О. Погоріляк. – Ужгород : ПП «АУТДОР-ШАРК», 2021. – 124 с.
4. Нікольський, Ю. В. Дискретна математика : підручник / Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина. – Львів : Магнолія, 2018. – 432 с.
5. Логічна алгебра: методичний посібник./ Н.А. Якімова. – Одеса: «Освіта України», 2019. – 40 с.
6. Математична логіка та теорія алгоритмів: Навчальний посібник/З.П. Халецька, В.В. Нарadowsкий. – Кропивницький: РВВ КДПУ ім. В. Винниченка, 2017. – 128 с.
7. Архітектура комп'ютерів. Частина 1 : лабораторний практикум / Л. В. Крупельницький, А. В. Снігур, С. В. Богомолів. – Вінниця : ВНТУ, 2020. – 104 с.
8. Архітектура комп'ютера та конфігурування комп'ютерних систем (на основі фундаменталізованого підходу) : навч. посіб. / Антоненко О. В., Бардус І. О. – Бердянськ : 2018 – 292 с.
9. Архітектура комп'ютерів та периферійні пристрої: Навч. посібник / С. Є. Бантюков, О. В. Чаленко, В. С. Меркулов та ін. – Харків: УкрДУЗТ, 2018. – Ч. 1. – 116 с.
10. Архітектура комп'ютера. Навчальний посібник./ Матвієнко М. П., Розен В. П., Закладний О. М. — К: Видавництво Ліра-К, 2016. — 264 с.
11. Зацеркляний М.М. Комп'ютерні основи систем кібербезпеки: навч. посібник/ Зацеркляний М.М., Струков В.М.-Харків: Тов. «В деле», 2017.- 292 с.

Додаткова література

1. Зацеркляний М.М. Інформаційні системи і технології в діяльності правоохоронних
2. Основи інформаційних технологій і систем- Підручник / В. А. Павлиш, Л. К. Гліненко, Н. Б. Шаховська. Львів : Видавництво Львівської політехніки, 2018. 620 с
3. Матвієнко М.П., Шаповалов С.П. Математична логіка та теорія алгоритмів. Навчальний посібник. – КК.:

Видавництво Ліра-К, 2017. – 212 с.

4. Матвієнко М.П., Шаповалов С.П. Математична логіка та теорія алгоритмів. Навчальний посібник. — Математичний практикум. — Київ : Ліра-К, 2015. — 212 с
5. Тарарака В.Д. Архітектура комп'ютерних систем: навчальний посібник. – Житомир :ЖДТУ, 2018. – 383 с.
6. Schwichtenberg, Helmut (2003–2004). *Mathematical Logic*. Munich, Germany:Mathematisches Institut der Universität München. Процитовано 2016-06-14. (англ.)

Інформаційні ресурси в Інтернеті

1. Математична логіка. Практикум [Електронний ресурс]: навч. посіб. для студ. спеціальності 113 «Прикладна математика», освітньої програми «Наука про дані та математичне моделювання» / О.Л.Темнікова ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,37 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 76 с.
<https://ela.kpi.ua/bitstream/123456789/42844/1/WorkshopLogicTemnikova.pdf>

Перелік програмного забезпечення

1. Операційна система MS Windows 7-10 - для засвоєння правил роботи з системою введення-виведення інформації та її зберігання на зовнішніх носіях.
2. Операційна оболонка (TotalCommander або аналогічна) - для засвоєння правил роботи з файлами, що зберігаються на носіях інформації.
3. Набір прикладних сервісних програм (антивірусні програми, архіватори).