Підручник з Okteta

Friedrich W. H. Kossebau
Alex Richardson
Переклад українською: Юрій Чорноіван

Зміст

Анотація

Okteta — це простий редактор для безпосереднього редагування даних файлів. Програми такого типу також називають шістнадцятковими або двійковими редакторами.

Розділ 1

Вступ

Okteta — це простий редактор для безпосереднього редагування даних файлів.

Дані буде показано у двох формах: як числові значення байтів та як символи, що відповідають цим числовим значенням. Значення та символи може бути показано або у два стовпчики (типовий формат перегляду у шістнадцяткових редакторах) або у форматі рядків, де значення буде показано над символом. Ви зможете змінювати як значення, так і символи.

Окрім звичайних засобів редагування, до складу Okteta входить також невеликий набір інструментів, зокрема інструмент декодування у прості типи даних, таблиця всіх байтів з символами і значеннями, інформаційне вікна зі статистичними даними, інструмент фільтрування та інструмент видобування рядків.

Всі зміни у завантажених даних можна без обмежень скасовувати та повторювати.

Розділ 2

Основи

2.1 Як запустити Okteta

Виконайте команду `okteta` з командного рядка або оберіть пункт **Двійковий редактор** з підменю **Програми** → **Інструменти** у інструменті запуску програм.

Можна скористатися стандартними параметрами командного рядка Qt™ і KDE Frameworks 5. Список цих параметрів можна переглянути у даних, виведених командою `okteta --help`.

Параметрами командного рядка, специфічними для Okteta є

`<URL>` — відкрити файли за вказаними адресами або адресою URL.

2.2 Використання

Головне вікно Okteta складається з таких частин: смужки меню, панелі інструментів, смужки стану, одної або декількох бічних панелей з інструментами та основної області перегляду з вкладками показу даних.

Після відкриття файла або створення нового масиву байтів з байтів, що у них містяться, буде побудовано послідовний список, кожен з рядків якого міститиме задану кількість байтів. Рядки буде показано у двох варіантах: як числові значення і як символи, що відповідають цим значенням. Значення і символи може бути показано або розділеними на дві колонки, або поряд одне з одним (значення буде показано над символом). Ліворуч від списку буде показано відступи першого байта кожного з рядків.

Робота з даними подібна до роботи у більшості текстових редакторів: дані можна змінювати, вирізати, копіювати, вставляти, перетягувати і скидати, точно так само, як ви це робите у текстовому редакторі. Курсор позначає поточну позицію. Натискання клавіші **Insert** перемикає режим редагування між перезаписом і вставкою. Режим перезапису працює трохи інакше, ніж у текстових редакторах, оскільки програма забороняє будь-які дії, які змінюють розмірність масиву байтів.

На відміну від текстових редакторів дані буде показано у двох варіантах. Одночасно для введення даних буде придатним лише один з цих варіантів. Програма показуватиме два пов'язані між собою курсори для значень і для символів, активний курсор бліматиме. Якщо буде активним поле символів, ви зможете вводити символи у спосіб, звичний для текстових редакторів. Якщо активним буде поле значень, введення цифри відкриватиму маленьке віконечко редактора, у якому ви зможете ввести решту значення.

Діалогове вікно пошуку надає користувачеві можливість пошуку вказаного рядка, байтів, які можна визначати у шістнадцятковому, десятковому, вісімковому, двійковому вигляді або у вигляді тексту (у поточному 8-бітовому кодуванні або UTF-8).

Одночасно можна відкривати декілька масивів даних, але лише один з них може бути активним. Щоб обрати активний масив байтів, скористайтеся меню **Вікна**.

Розділ 3

Інструменти

3.1 Огляд

У Okteta передбачено декілька інструментів. Деякі з них відповідають за аналіз та обробку масивів даних, деякі мають загальне призначення. Задіяти або вимкнути ці інструменти можна за допомогою меню **Інструменти**. Кожен з інструментів має власну невеличку панель, яку буде розташовано на одній з бічних панелей або у окремому віконці. Ви можете швартувати, від'єднувати, перевпорядковувати та розташовувати стосами панельки інструментів за допомогою миші: наведіть вказівник миші на смужку заголовка панелі інструменту, натисніть ліву кнопку миші, пересуньте панель у бажане місце і відпустіть кнопку, щоб завершити компонування, або клавішу **Esc**, щоб скасувати пересування панелі.

3.1.1 Інструменти аналізу та обробки

Таблиця значень/символів

У цій таблиці показано список можливих значень байтів у вигляді символів і знаків у різних числових кодуваннях.

Обране значення можна вставляти за поточним розташуванням курсора визначену кількість разів. Дію з вставлення можна виконати за допомогою кнопки Вставити або наведення вказівника на відповідний рядок таблиці з наступним подвійним клацанням.

Двійковий фільтр

Фільтри виконує двійкові дії над вибраними байтами. Після вибору дії (AND, OR, ROTATE..) параметри, якщо такі є, можна встановити у полі, розташованому нижче. Виконати фільтрування можна за допомогою кнопки Фільтрувати.

Рядки

За допомогою цього інструмента можна шукати рядки у вибраних байтах. Після того, як ви оберете мінімальну довжину рядка, рядки можна знайти за допомогою кнопки **Видобути**. Список рядків можна звузити введенням шаблону фільтрування.

Статистика

За допомогою цього інструменту можна зібрати деяку статистичну інформацію щодо вибраних байтів. Серед цих даних будуть дані щодо частоти входження кожного з байтових значень до виділеного фрагмента. Наказати програмі виконати збирання інформації можна натисканням кнопки Зібрати.

Контрольна сума

За допомогою цього інструменту можна обчислювати різноманітні контрольні суми та хеш-суми для позначених байтів. Після вибору алгоритму та встановлення значення параметра, якщо такий потрібен, відповідну суму можна обчислити натисканням кнопки **Обчислити**.

Таблиця розкодування

У цій таблиці показано значення байта або байтів, починаючи від поточного розташування курсора, як їх відповідники серед простих типів даних, зокрема цілих чисел (Integer) або чисел з плаваючою комою (Float), але також у форматі UTF-8. Якщо ви наведете вказівник миші на рядок у таблиці і двічі клацнете лівою кнопкою миші, програма відкриє вікно редактора, у якому ви зможете переглянути і змінити значення.

Структури

За допомогою цього інструменту можна вивчати та редагувати масиви байтів на основі визначених користувачем структур. Докладніший опис наведено у окремому розділі.

3.1.2 Загальні інструменти

Файлова система

Панель цього інструменту є вбудованим навігатором файловою системою, яким можна скористатися для позначення файлів, які слід відкрити.

Документи

На панелі цього інструменту буде показано пункти всіх поточних створених або завантажених файлів. Програма позначить відповідним чином пункти файлів, перегляд яких є активним, файли з незбереженими змінами та файли, які було змінено на носії даних іншою програмою.

Закладки

Панеллю цього інструменту можна скористатися для керування закладками, якщо ви не бажаєте користуватися меню **Закладки**.

> Примітка
> У поточній версії програми закладки є лише тимчасовими, програма не зберігає їх розташування під час закриття масиву байтів або завершення роботи самої програми.

Інформація про файл

За допомогою цього інструменту можна переглянути відомості щодо поточного файла, зокрема його тип, місце зберігання і розмір.

Термінал

Вбудований термінал. Робочий каталог не синхронізовано з каталогом активного файла.

Перетворення наборів символів

За допомогою цього інструмента можна перезаписати відповідні символи у іншому кодуванні. Передбачено підтримку лише 8-бітового набору символів. У поточній версії усі символи без відповідників буде замінено на 0.

3.2 Інструмент структур

3.2.1 Загальне

За допомогою інструменту «Структури» можна аналізувати та редагувати масиви байтів на основі створених користувачем визначень структур, які можна побудувати з масивів, об'єднань, елементарних типів та переліків.

Інструмент має власне діалогове вікно параметрів, відкрити яке можна за допомогою натискання кнопки **Параметри**. Передбачено досить багато параметрів налаштування, зокрема стиль (десяткові, шістнадцяткові або двійкові), у якому буде показано значення. Крім того, можна визначити, які визначення структур буде завантажено і які структури буде показано.

Структури визначаються за допомогою файлів визначення структур Okteta (XML-файлах, з суфіксом назви `.osd`). Крім того, потрібен файл `.desktop`, що містить метадані щодо файла опису структури, зокрема відомості щодо автора, домашньої сторінки та умов ліцензування.

У поточній версії програми ще немає вбудованої підтримки створення або редагування визначень структур, отже все це слід робити вручну у спосіб, описаний у наступних розділах.

3.2.2 Встановлення визначень структур

3.2.2.1 Встановлення за допомогою KNewStuff

Найпростішим способом встановлення нових визначень структур є використання вбудованої підтримки KNewStuff у Okteta. Щоб встановити існуючу структуру, відкрийте діалогове вікно інструменту «Структури». Перейдіть на вкладку **Керування структурами** і натисніть кнопку **Отримати нові структури...**. За допомогою діалогового вікна, яке буде відкрито, ви зможете встановити або вилучити встановлені визначення структур.

3.2.2.2 Встановлення визначень структур вручну

Інструмент «Структури» шукає описи структур у підкаталозі `okteta/structures/` каталогу даних програм користувача (визначити адресу цього каталогу можна за допомогою команди `qtpaths --paths GenericDataLocation`). Якщо ще не встановлено жодного визначення структури, можливо, вам доведеться створити відповідний підкаталог.

Дані кожного визначення структури має бути розподілено між двома файлами: один з файлів міститиме дані самого визначення, а інший файл, `.desktop`, зберігатиме метадані (дані щодо автора, версії тощо).

У цьому каталозі має бути підкаталог для кожного з визначень структури, у якому мають зберігатися обидва файла визначення: файл `.desktop` і файл `.osd` або `main.js`.

Наприклад, якщо каталогом даних програм є `qtpaths --paths GenericDataLocation`, а визначення структури має назву ExampleStructure, має бути каталог `okteta/structures/ExampleStructure`, що міститиме файли `ExampleStructure.desktop` і `ExampleStructure.osd`.

3.2.2.3 Користування встановленими структурами

Якщо ви встановили нове визначення структури створенням такого каталогу або редагуванням файлів у ньому, вам слід перезапустити Okteta, а потім відкрити діалогове вікно параметрів інструменту «Структури». Там слід перейти на вкладку **Керування записами структур** і переконатися, що відповідний пункт визначення структури позначено. Після цього перейдіть на вкладку **Структури** і переконайтеся, що бажаний пункт є у списку, розташованому у правій частині вікна.

3.2.3 Поширення визначень структур

Якщо потрібна вам структура є досить типовою, ви можете не створювати визначення власноруч, а скористатися готовим визначенням з якогось зі сховищ визначень, зокрема store.kde.org.

Крім того, ви і самі можете поділитися з іншими користувачами вашими визначеннями. Щоб зробити це, створіть файл архіву (наприклад, стиснутий zip архів tar, `.tar.gz`), який міститиме лише підкаталог з файлом `.desktop` і файлом визначення структури. Якщо скористатися нашим прикладом з попереднього розділу, ваш підкаталог має називатися `ExampleStructure`. Якщо ви користуватиметеся саме цим форматом інші користувачі зможуть автоматично отримувати і встановлювати створені вами визначення.

3.2.4 Створення визначень структур

> **Примітка**
> Новіші, але трохи неповні настанови щодо створення визначень структур можна знайти у вікі користувачів KDE.

Передбачено два різних способи створення визначень структури. Перший полягає у написанні визначення мовою XML, інший — мовою JavaScript. За допомогою JavaScript можна створювати складніші описи структур, зокрема з перевірками коректності структури. Використання XML звузить набір можливостей, але якщо вам потрібна лише статична структура, цей спосіб буде простішим. Якщо вам слід описати динамічну структуру, наприклад структуру, у якій довжини масивів залежатимуть від інших даних у структурі, або структуру, компонування якої змінюватиметься залежно від значень її компонентів, вам доведеться описувати структуру мовою JavaScript. Єдиним виключенням з цього правила є випадок, коли довжина масиву **точно** дорівнює іншому значенню структури. У цьому випадку ви можете скористатися XML, але якщо ви маєте для довжини щось подібне до *значення - 1*, доведеться користуватися JavaScript.

3.2.5 Формат файлів визначення структури XML

> **Примітка**
> Новіші, але трохи неповні настанови щодо створення визначень структур можна знайти у вікі користувачів KDE.

У файла XMLe `.osd` є один кореневий теґ: <*data*> без жодних атрибутів. Всередині цього теґу має бути один з таких теґів:

<primitive>

Для створення елементарних типів даних, наприклад *int* або *float*. Цей теґ не містить підтеґів і може мати такі атрибути:

type

Тип цього елементарного типу. Має бути одним з таких значень:

- *char* для 8-бітових символів ASCII
- *int8, int16, int32, int64* для цілих значень відповідного розміру зі знаком
- *uint8, uint16, uint32, uint64* для додатних цілих чисел відповідного розміру
- *bool8, bool16, bool32, bool64* для додатних булевих значень (0 = false, всі інші значення = true) відповідного розміру
- *float* для 32-бітових дійсних значень стандарту IEEE754
- *double* для 64-бітових дійсних значень стандарту IEEE754

<bitfield>

Для створення бітового поля. Цей елемент не містить піделементів і може мати такі атрибути:

width

Кількість бітів, використаних відповідним бітовим полем. Повинен мати значення від 1 до 64.

type

Тип бітового поля. Має бути одним з таких записів:

- *unsigned* для бітових полів, значення яких слід вважати додатним (зі значенням у діапазоні від 0 до 2^{width} - 1)

- *signed* для бітових полів, значення яких слід вважати значенням зі знаком (зі значенням від -$2^{\text{width - 1}}$ до $2^{\text{width - 1}}$ - 1)
- *bool* для бітових полів з булевим значенням

> Примітка
> Не забувайте додавати символи розриву рядка після *<bitfield>*, інакше наступний елемент (окрім рядків та масивів, там розрив буде додано автоматично) розпочнеться посередині байта. Звичайно ж, додавання розривів непотрібне, якщо саме така поведінка і є бажаною.

<enum>

Для створення елементарного типу, значення якого буде, якщо це можливо, показано як нумерований список. Для цього теґу не передбачено підтеґів (але вам потрібен буде теґ *<enumDef>* у файлі для посилання). Має такі атрибути:

enum
: Нумерований список — основа цього значення. Маж збігатися зі значенням *name* одного з теґів *<enumDef>* у цьому файлі.

type
: Тип цього нумерованого списку. Див. атрибут type *<primitive>*. Єдиною відмінністю є те, що значення *Double* і *Float* у цьому випадку позбавлені сенсу.

<flags>

Те саме, що і *<enum>*, єдиною відмінністю є те, що значення представлено як *побітове «або»* для всіх значень нумерованого списку.

<struct>

Для створення структури. Всі інші теґи (зокрема *<struct>*) можуть бути дочірніми теґами такої структури, отже будуть частиною отриманої структури.

<union>

Для створення об'єднання. В основному, те саме, що і *<struct>*, окрім того, що всі дочірні елементи будуть починатися з однаковим відступом. Корисно для представлення однієї послідовності байтів у різний спосіб.

<array>

Для створення масиву. У цього теґу має бути один і лише один дочірній теґ (тип елемента масиву), який може визначати будь-який теґ, навіть теґ *<array>*. Також має такі атрибути:

length
: Кількість елементів у цьому масиві у форматі десяткового числа. Крім того, може мати значення, яке збігається з назвою атрибута попередньо визначеного теґу *<primitive>*, *<enum>* або *<flags>*. У такому разі значенням length вважатиметься значення відповідного теґу. У поточній версії значення обмежено числом 10000, оскільки більші значення призводять до надмірного споживання пам'яті і значно уповільнюють роботу програми.

<string>

Для створення рядка у певному кодуванні. З атипових значень буде визначено рядок у стилі C, що завершується символом *NULL*. Інші типи рядків може бути створено за допомогою таких атрибутів:

terminatedBy
: Це атрибут визначає символ unicode, яким має завершуватися рядок. Символ слід вказувати у форматі шістнадцяткового номер у таблиці (з можливим додаванням префікса *0x*). Якщо встановлено кодування ASCII, можна використовувати лише значення до 0x7f. Якщо не встановлено ні цього атрибута, ні атрибутів *maxCharCount* і *maxByteCount*, вважатиметься, що значенням є 0 (рядок у стилі C).

maxCharCount
: Максимальна кількість символів у рядку. Якщо встановлено також атрибут *terminatedBy*, рядок буде обірвано за виконання будь-якої з умов. Не можна визначати разом з атрибутом *maxByteCount*.

maxByteCount
: Максимальна кількість байтів у рядку. Якщо встановлено також атрибут *terminatedBy*, рядок буде обірвано за виконання будь-якої з умов. Не можна визначати разом з атрибутом *maxCharCount*. Для кодувань, подібних до *ASCII*, збігається з *maxCharCount*.

type
: Кодування цього рядка. Може приймати такі значення:
  - *ASCII*
  - *LATIN-1*
  - *UTF-8*
  - *UTF-16-LE* або *UTF-16-BE*. Якщо не вказано суфікса *-LE* або *-BE*, вважатиметься, що кодуванням є *-LE* (little endian).
  - *UTF-32-LE* або *UTF-32-BE*. Якщо не вказано суфікса *-LE* або *-BE*, вважатиметься, що кодуванням є *-LE* (little endian).

Кожен з елементів містить атрибут *name*, значення якого буде показано на панелі перегляду структури.

3.2.6 Зразок визначення структури мовами XML і JavaScript

> Примітка
> Новіші, але трохи неповні настанови щодо створення визначень структур можна знайти у вікі користувачів KDE.

3.2.6.1 Типовий крок, який буде спільним для обох підходів:

Наш файл метаданих буде таким:

```
[Desktop Entry]
Icon=arrow-up<:\coref{1}{icon}:>
Type=Service
ServiceTypes=KPluginInfo

Name=Simple test structure
Comment=A very simple test structure containing only two items

X-KDE-PluginInfo-Author=Alex Richardson
X-KDE-PluginInfo-Email=foo.bar@email.org
X-KDE-PluginInfo-Name=simplestruct
X-KDE-PluginInfo-Version=1.0
X-KDE-PluginInfo-Website=https://www.plugin.org/
X-KDE-PluginInfo-Category=structure
X-KDE-PluginInfo-License=LGPL
X-KDE-PluginInfo-EnabledByDefault=false
```

❶ Піктограма, яку буде показано Okteta для відповідного пункту структури. Можна вказати будь-яку піктограму, яку можна знайти у вікні, яке відкриває команда `kdialog --geticon`, або адресу файла піктограми

Призначення полів має бути очевидним, окрім поля `X-KDE-PluginInfo-Name`. Значення у цьому полі має збігатися з назвою каталогу, у якому зберігається файл, а також назвою файла `.desktop`. Під час створення структури XML назва файла `.osd` також має збігатися з вказаною назвою.

У нашому прикладі ми маємо каталог з назвою `simplestruct`, у якому зберігаються файл `simplestruct.desktop`. Якщо структура визначається у форматі XML, у каталозі також має зберігатися файл з назвою `simplestruct.osd`. Якщо використано підхід з визначенням JavaScript, матимемо файл з назвою `main.js`.

3.2.6.2 Просте визначення структури у форматі XML

Для початку ми створимо визначення дуже простої структури, що міститиме лише прості типи даних (один символ і одне 32-бітове ціле число (int)). Цю структуру можна записати мовою C/C++ так:

```
struct simple {
  char aChar;
  int anInt;
  bool bitFlag :1;
  unsigned padding :7;
};
```

Першим кроком буде створення файла `.osd`, який ми назвемо `simplestruct.osd`:

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
  <struct name="simple">
    <primitive name="aChar" type="Char"/>
    <primitive name="anInt" type="Int32"/>
    <bitfield name="bitFlag" type="bool" width="1"/>
    <bitfield name="padding" type="unsigned" width="7"/>
  </struct>
</data>
```

Цей файл дуже подібний до визначення структури у C/C++.

Тепер створимо підкаталог `simplestruct` у каталозі встановлення структури (див. встановлення визначення структури вручну) і скопіюємо два файли до цього каталогу. Тепер можна перезапустити Okteta і спробувати скористатися новою структурою.

3.2.6.3 Проста структура мовою JavaScript

Щоб реалізувати наведену вище структуру мовою JavaScript, створіть файл з назвою `main.js` замість `simplestruct.osd`, і замініть X-KDE-PluginInfo-Category=structure на X-KDE-PluginInfo-Category=structure/js. У файлі мають міститися такі рядки:

```
function init() {
  var structure = struct({
    aChar : char(),
    anInt : int32(),
    bitFlag : bitfield("bool", 1),
    padding : bitfield("unsigned", 7),
  })
  return structure;
}
```

Структура, показана у вікні Okteta, це завжди значення, яке повертає функція `init`.

Для створення типу елемента може бути викликано такі функції:

- char()
- int8(), int16(), int32() or int64()
- uint8(), uint16(), uint32() or uint64()
- bool8(), bool16(), bool32() or bool64()
- float()
- double()

Функція bitfield отримує два параметри. Першим є рядок `bool`, `signed` або `unsigned`. Другим параметром є ціле число, яке встановлює розмір типу у бітах.

3.2.6.4 Складніші структури

Тепер ми створимо визначення складнішої структури, яке назвемо «complex» і збережемо у файлі з назвою `complex.osd`. Ця структура міститиме два масиви (один фіксованої довжини і один, у якому довжина визначатиметься під час роботи програми), а також вкладену структуру і об'єднання (union).

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
  <struct name="complex">
    <primitive name="size" type="UInt8" />
    <union name="aUnion">
      <array name="fourBytes" length="4">
        <primitive type="Int8" />
      </array>
    </union>
    <struct name="nested">
      <array name="string" length="size"> <!-- references the  ↩
          field size above -->
        <primitive type="Char" />
      </array>
    </struct>
  </struct>
</data>
```

Така структура формально визначається у C/C++ так:

```
struct complex {
    uint8_t size;
    union aUnion {
      int8_t fourBytes[4];
    };
    struct nested {
      char string[size] //не відповідає стандартам C++, посилает ↩
          ься на значення розмірності uint8
    };
};
```

> **Примітка**
> Очевидно, поля посилань на масиви змінної (динамічної) довжини слід вказувати до самого масиву.

Далі, ми створимо файл `complex.desktop`, як і у попередньому прикладі (переконайтеся, що значення `X-KDE-PluginInfo-Name` змінено відповідним чином), та повторимо дії зі встановлення визначення структури.

3.2.6.5 Докладніші відомості

Декілька прикладів визначень структур можна знайти тут. Серед цих прикладів є файл заголовка для файлів PNG і заголовка файла ELF. Схема XML, яка описує структуру файла `.osd`, зберігається тут. Якщо вам потрібні якісь додаткові відомості, зверніться до автора за адресою arichardson.kde@gmail.com

Розділ 4

Огляд інтерфейсу

4.1 Пункти меню

Окрім типових для KDE пунктів меню, описаних у розділі щодо меню підручника з основ роботи у KDE, у Okteta передбачено такі додаткові пункти:

4.1.1 Меню «Файл»

Файл → Створити (Ctrl+N)

Створити масив байтів...

- **Порожній**: ...як порожній.
- **З буфера обміну даними**: ...як поточний вміст буфера обміну даними.
- **Шаблон...**: ...за вказаним шаблоном.
- **Випадкові дані**: ...заповненим випадковими даними.
- **Послідовність**: ...з усіма байтами від 0 до 255.

Файл → Експортувати

Експортувати позначені байти до файла так, щоб їх було...

- **Значення**: ...закодовано як байтові значення. Типово значення буде відокремлено одним символом пробілу. Символи **Відокремлення** можна змінити за допомогою діалогового вікна **Експортувати**.
- **Символи**: ...закодовано як звичайний текст.
- **Base64**: ...закодовано у форматі Base64.
- **Base32**: ...закодовано у форматі Base32.
- **Ascii85**: ...закодовано у форматі Ascii85.
- **Uuencoding**: ...закодовано у форматі Uuencoding.
- **Xxencoding**: ...закодовано у форматі Xxencoding.
- **Intel Hex**: ...закодовано у форматі Intel Hex.
- **S-Record**: ...закодовано у форматі S-Record.
- **Масив C**: ...у вигляді масиву, що використовується у мові програмування C.
- **Переглянути як звичайний текст**: ...як у перегляді даних з відступом, значення байтів і символи.

Файл → Дозволи → Лише для читання

Встановлення такого дозволу забороняє вносити зміни до завантаженого масиву байтів.

Файл → Закрити всі інші

Закрити всі масиви байтів, окрім поточного.

4.1.2 Меню «Зміни»

Зміни → Копіювати як

Скопіювати позначені байти у одному з форматів до буфера обміну даними. Перелік доступних форматів наведено у описі пункту меню **Файл → Експортувати**

Зміни → Вставити

Вставити шаблон...

Вставляє вказаний рядок байтів за розташування курсора.

Параметри діалогового вікна нададуть вам змогу вказати кількість вставлень шаблону і його формат (шістнадцятковий, десятковий, вісімковий, двійковий, символи або UTF-8).

Зміни → Скасувати вибір (Ctrl+Shift+A)

Скасувати поточний вибір.

Зміни → Вибрати діапазон… (Ctrl+E)

Відкрити вбудоване діалогове вікно для введення діапазону позначення.

Зміни → Режим перезапису (Ins)

Перемикає режим редагування з режиму вставки на режим перезапису.

> Примітка
> Режим перезапису реалізовано у дуже строгому варіанті: не можна змінювати розмір даних (додавати або вилучати байти).

Зміни → Пошук... (Ctrl+F)

Шукає вказаний шаблон у документі. Можна шукати шістнадцяткові, десяткові, вісімкові, двійкові, двійкові або текстові шаблони.

Параметри відповідного діалогового вікна нададуть вам змогу вказати початкову точку, напрям і діапазон пошуку.

Зміни → Зсув переходу... (Ctrl+G)

Пересуває курсор до вказаного зсуву у байтах.

4.1.3 Меню «Перегляд»

Перегляд → Показати відступ рядка (F11)

Вмикає або вимикає показ зсуву рядка на панелі, розташованій ліворуч.

Перегляд → Показувати значення або символи

Цей пункт надає вам змогу обрати, які інтерпретації байтів буде показано. Можливі варіанти:

- **Значення**
- **Символи**
- **Значення і символи**

Перегляд → Кодування значення

Надає змогу обрати кодування значень з таких варіантів:

- **шістнадцяткове**
- **десяткове**
- **вісімкове**

- **двійкове**

Перегляд → Кодування символів

За допомогою цього підменю ви можете обрати кодування символів.

Перегляд → Показувати недруковані символи

Вмикає або вимикає показ недрукованих символів. Якщо показ вимкнено, на відповідних місцях у стовпчику символів замість недрукованих символів буде показано символ-замінник.

Перегляд → Встановити кількість байтів на рядок

Вибрати кількість показаних у рядку байтів. Типово буде показано 16 байтів.

Перегляд → Встановити кількість байтів на групу

Типово шістнадцяткові значення буде показано групами по 4 байти. За допомогою цього пункту меню ви можете скоригувати параметри показу.

Перегляд → Динамічне компонування

Встановлює правила компонування показу даних. Тут можна визначити скільки байтів буде показано у рядку, залежно від ширини області перегляду. Можливі такі правила:

- **Вимкнути**: буде зафіксовано поточну кількість байтів на рядок, яке не змінюватиметься після зміни розмірів області перегляду.
- **Переносити лише повні групи байтів**: вивести найбільшу можливу кількість байтів у рядку, але число байтів у групі має бути цілим.
- **Увімкнути**: те саме, що і попередній пункт, але дозволяє використання груп байтів з нецілого числа байтів.

Перегляд → Режим перегляду

Надає змогу обрати компонування перегляду з таких варіантів:

- **Стовпчики**: інтерпретацію значень і символів буде показано у класичному компонуванні: кожен список у окремому стовпчику.
- **Рядки**: символьне тлумачення байта буде показано безпосередньо під інтерпретацією у вигляді значення.

Зміни → Розділити горизонтально (Ctrl+Shift+T)

Розділити область поточного фокусованого перегляду на дві частини і додати копію поточної області до нової, розташованої нижче, області.

Зміни → Розділити вертикально (Ctrl+Shift+L)

Розділити область поточного фокусованого перегляду на дві частини і додати копію поточної області до нової, розташованої праворуч, області.

Перегляд → Закрити область перегляду (Ctrl+Shift+R)

Закрити поточну фокусовану область перегляду.

Перегляд → Профіль перегляду

Параметри перегляду можна зберігати у профілях перегляду. Оновити поточний вибраний профіль можна безпосередньо за допомогою параметрів поточного перегляду або за допомогою створення нового профілю на основі поточного. Керувати профілями перегляду можна за допомогою діалогового вікна, яке відкривається пунктом меню **Параметри → Керування профілями перегляду...**.

4.1.4 Меню Вікна

У цьому меню буде показано список поточних областей перегляду. Тут можна обрати активне вікно.

4.1.5 Меню Закладки

У одному масиві байтів можна встановити декілька закладок. Кожен масив байтів має свій набір закладок, список відповідних закладок буде показано внизу меню **Закладки**. Щоб пересунути курсор до закладки і переглянути байти за нею, оберіть відповідну закладку у цьому меню.

> Примітка
> У поточній версії програми закладки є лише тимчасовими, програма не зберігає їх розташування під час закриття масиву байтів або завершення роботи самої програми.

Закладки → Додати закладку (Ctrl+B)

Позначити місце у масиві байтів закладкою.

Закладки → Вилучити закладку (Ctrl+Shift+B)

Вилучає поточну закладку. Доступ до цієї команди можна отримати, лише якщо курсор знаходиться у місці, на яке встановлено закладку.

Закладки → Вилучити всі закладки

Спорожнює список закладок.

Закладки → Перейти до попередньої закладки (Alt+Up)

Пересуває курсор до попередньої закладки.

Закладки → Перейти до наступної закладки (Alt+Down)

Пересуває курсор до наступної закладки.

4.1.6 Меню «Інструменти»

Показує список встановлених інструментів. У цьому списку можна увімкнути або вимкнути показ будь-якого з цих інструментів. Докладний опис всіх інструментів наведено у розділі Інструменти.

4.1.7 Меню «Параметри»

Параметри → Керування профілями перегляду...

Відкрити діалогове вікно для створення, редагування, вилучення та встановлення типового профілю перегляду.

Розділ 5

Подяки і ліцензія

Okteta

Переклад українською: Юрій Чорноіван yurchor@ukr.net