

# Приручник за Октету

Фридрих В.Х. Косебау  
Александер Ричардсон  
превод: Драган Пантелић



## Приручник за Октету

# Садржај

<b>1</b>	<b>Увод</b>	<b>5</b>
<b>2</b>	<b>Основе</b>	<b>6</b>
2.1	Покретање Октеге . . . . .	6
2.2	Коришћење . . . . .	6
<b>3</b>	<b>Алатке</b>	<b>7</b>
3.1	Преглед . . . . .	7
3.1.1	Анализатори и манипулатори . . . . .	7
3.1.2	Опште алатке . . . . .	8
3.2	Алатка структура . . . . .	8
3.2.1	Опште . . . . .	8
3.2.2	Инсталирање дефиниција структура . . . . .	9
3.2.2.1	Инсталирање преко врућих новотарија . . . . .	9
3.2.2.2	Ручно инсталирање дефиниција структура . . . . .	9
3.2.2.3	Коришћење инсталираних структура . . . . .	9
3.2.3	Дељење дефиниција структура . . . . .	9
3.2.4	Стварање дефиниција структура . . . . .	9
3.2.5	ИксМЛ формат фајла за дефинисање структура . . . . .	10
3.2.6	Пример исте структуре у ИксМЛ-у и јаваскрипту . . . . .	12
3.2.6.1	Заједнички кораци за оба приступа . . . . .	12
3.2.6.2	Једноставна ИксМЛ дефиниција структуре . . . . .	12
3.2.6.3	Једноставна структура у јаваскрипту . . . . .	13
3.2.6.4	Сложеније структуре . . . . .	13
3.2.6.5	Још информација . . . . .	14
<b>4</b>	<b>Преглед сучеља</b>	<b>15</b>
4.1	Ставке менија . . . . .	15
4.1.1	Мени <b>Фајл</b> . . . . .	15
4.1.2	Мени <b>Уређивање</b> . . . . .	16
4.1.3	Мени <b>Приказ</b> . . . . .	16
4.1.4	Мени <b>Прозори</b> . . . . .	17
4.1.5	Мени <b>Обележивачи</b> . . . . .	18
4.1.6	Мени <b>Алатке</b> . . . . .	18
4.1.7	Мени <b>Поставке</b> . . . . .	18
<b>5</b>	<b>Заслуге и лиценца</b>	<b>19</b>

### **Сажетак**

Октета је једноставан уређивач за сирове податке у фајловима. Овај тип програм такође се назива хексадекадним или бинарним уређивачем.

# Глава 1

## Увод

Октета је једноставан уређивач за сирове податке у фајловима.

Подаци су приказани у две варијанте: као бројевне вредности бајтова, и као знакови придружени тим вредностима. Вредности и знакови могу бити приказани или у две колоне (традиционални приказ у хексадекадним уређивачима) или по врстама са вредношћу изнад знака. Уређивати се може и по вредностима и по знаковима.

Поред уобичајених уређивачких способности, Октеда пружа и мали скуп специјалних алатки, као што су табеларно набрајање декодирања према уобичајеним типовима података, табеларно набрајање свих могућих бајтова са знаковним и вредносним еквивалентима, информативни приказ статистике, калкулатор контролних сума, алатка за филтрирање и алатка за издвајање ниски.

Све измене над подацима могу бити опозиване и понављане без ограничења.

## Глава 2

# ОСНОВЕ

### 2.1 Покретање Октете

Упишите `okteta` у командни одзивник или изаберите **Хексадекадни уређивач** из групе **Програми** → **Алатке** у покретачу програма.

Доступне су стандардне `КуТ` и `КФ5` опције командне линије, а набројати се могу извршавањем `okteta --help`.

Опције командне линије посебне за Октеду су:

`<УРЛ-ови>` — отвара фајлове са задатих УРЛ-ова.

### 2.2 Коришћење

Главни прозор Октете има следеће компоненте: траку менија, траку алатки, траку стања, једну или више бочних трака са алаткама, и главни приказ података са језичцима.

Када се отвори фајл или направи нови низ бајтова, садржај бајтова се ниже у редове са задатим бројем бајтова по реду. Приказују се у два облика: као бројевне вредности и као знакови придружени тим вредностима. Вредности и знакови могу бити приказани или раздвојено у две колоне, или једни до других са вредношћу изнад знака. На левој страни дати су помаци првог бајта у сваком реду.

Ради се слично као у већини уређивача текста: подаци се могу уређивати, исецати, копирати, налепљивати и превлачити, као и обичан текст. Курсор означава тренутни положај. Притиском на тастер **Insert** мењате између режима пребрисавања и уметања. Режим пребрисавања је строжији него у уређивачима текста, у томе што не дозвољава поступке који би променили величину низа бајтова.

За разлику од уређивача текста, садржај се приказује у два облика; само један од њих може бити активан за прихватање улаза. Дата су два повезана курсора, један у приказу вредности а други у приказу знакова, и курсор активног приказа трепће. Када су активни знакови, притиском на прву цифру отвара се минимални уређивач ради уношења остатка вредности.

Дијалог за претрагу омогућава тражење одређене ниске бајтова, дефинисане као хексадекадна, декадна, октална или бинарна вредност, или као текст (по текућем кодирању или УТФ-8).

Истовремено може бити отворено више низова бајтова, али само један може бити активан. Активни низ бирате кроз мени **Прозори**.

## Глава 3

# Алатке

### 3.1 Преглед

Октета садржи разне алатке, неке за анализирање и манипулисање низовима бајтова, а неке са општијом наменом. Ове алатке се могу активирати и деактивирати преко ставке **Алатке** на траци менија. Свака алатка има свој мали приказ, који или пристаје у једну од бочних трака или слободно плута као прозор. Приказе алатки можете сидрити, откачињати, прераспоређивати, а такође и слагати тако што притиснете лево дугме миша над насловном траком приказа, померите га колико желите, па пустити дугме да довршите радњу (или тастер **Esc** да одустанете).

#### 3.1.1 Анализатори и манипулатори

##### Табела вредност—знак

Табела набраја све могуће вредности бајта, и нумеричке и знаковне у различитим кодирањима.

Изабрана вредност може се уметнути на положају курсора за дефинисани број бајтова. Умеће се дугметом **Уметни** или двокликом на ред у табели.

##### Бинарни филтер

Филтер изводи бинарне операције над изабраним бајтовима. Након избора операције (**И**, **ИЛИ**, **РОТИРАЈ...**), параметри се, ако их има, могу поставити у пољу испод. Филтер се примењује дугметом **Филтрирај**.

##### Ниске

Ова алатка проналази ниске у изабраним бајтовима. Након избора минималне дужине, тражење се покреће дугметом **Извучи**. Списак приказаних ниски може се сузити уношењем филтерског израза.

##### Статистика

Ова алатка гради статистику за изабране бајтове, коју чине учесталости понављања за сваку вредност бајта у избору. Може се прорачунати дугметом **Изгради**.

##### Контролна сума

Ова алатка рачуна разне контролне и дисперзионе суме изабраних бајтова. Пошто изаберете алгоритам и поставите параметар, ако га има, сума се израчунава притиском на дугме **Израчунај**.

##### Табела декодирања

Табела приказује вредности бајтова на положају курсора, за неке уобичајене једноставне типове података као што су целобројни или у покретном зарезу, али и UTF-8. Двоклик на врсту у табели отвара уређивач у којем се вредност може изменити.

## Структуре

Ова алатка омогућава испитивање и уређивање низова бајтова на основу кориснички задатих дефиниција структура. Детаљне инструкције дате су у [посебном одељку](#).

### 3.1.2 Опште алатке

#### Фајл систем

Ова алатка је угнежђени прегледач фајлова којим се могу бирати фајлови за отварање.

#### Документи

Ова алатка показује све тренутно учитане и направљене фајлове. Символима су означени фајл са тренутно активни приказом, фајлови који имају несачуваних измена и фајлови чију ускладиштену копију је изменио неки други програм.

#### Обележивачи

Ова алатка служи за управљање обележивачима, као алтернатива менију **Обележивачи**.

#### ПРИМЕДБА

Обележивачи су за сада пролазни, тј. не уписују се када затворите низ бајтова или цео програм.

#### Подаци о фајлу

Ова алатка наводи неке податке о текућем фајлу, као што су тип, локација складиштења и величина.

#### Терминал

Угнежђени терминал, радна фасцикла није повезана са активним фајлом.

#### Претварање кодирања

Претварање бајтова тако да записани знакови буду исти у другом кодирању. Подржана су само осмобитна кодирања, а непоклопљени знакови тренутно се замењују укодираном нулом.

## 3.2 Алатка структура

### 3.2.1 Опште

Алатка структура омогућава анализирање и уређивање низова бајтова на основу кориснички задатих дефиниција структура, које се могу састојати од низова, унија, примитивних типова и набројивих вредности.

Поседује сопствени дијалог поставки, који се позива дугметом **Поставке**. Могу се подесити разне опције, попут стила приказа вредности (декадно, хексадекадно, бинарно). Могуће је изабрати које се дефиниције структура читавају и које структуре се виде у приказу.

Структура се дефинише у фајлу Октетине дефиниције структуре (ИксМЛ формат са наставком `.osd`). На ово се додаје и `.desktop` фајл са метаподацима о фајлу описа структуре, попут аутора, домаће странице и лиценце.

Тренутно нема уграђене подршке за стварање и уређивање дефиниција структура, већ се то мора ручно радити како је описано у наредним одељцима.



## 3.2.2 Инсталирање дефиниција структура

### 3.2.2.1 Инсталирање преко врућих новотарија

Нове дефиниције структура најлакше се инсталирају путем подршке за вруће новотарије уграђене у Октеу. Да бисте инсталирали постојећу структуру, отворите дијалог поставки алатке за структуре. У њему идите под језичак **Управљање структурама** и кликните на дугме **Добави нове структуре....** Дијалог који се на то појави омогућава вам инсталирање и деинсталирање структура.

### 3.2.2.2 Ручно инсталирање дефиниција структура

Алатка структура тражи описе структура у потфасцикли `okteta/structures/` корисничке фасцикле за податке програмâ (коју налазите извршавањем `qtpaths --paths GenericDataLocation`). Можда ћете морати да направите ову фасциклу, ако још увек нема ниједне инсталиране дефиниције структуре.

За сваку дефиницију структуре постоје два фајла: један за саму дефиницију, и један `.desktop` фајл за метаподатке (аутор, верзија, итд.).

У наведеној фасцикли, свака дефиниција структуре има своју потфасциклу која садржи и `.desktop` и `.osd` или `main.js` фајл дефиниције.

На пример, ако је фасцикла података програмâ `qtpaths --paths GenericDataLocation` а име дефиниције структуре `PrimerStruktur`, онда постоји фасцикла `okteta/structures/ExampleStructure` која садржи фајлове `PrimerStruktur.desktop` и `PrimerStruktur.osd`.

### 3.2.2.3 Коришћење инсталираних структура

Пошто инсталирате нову дефиницију структуре, да бисте је користили морате поново покренути Октеу. Пошто то учините, отворите дијалог поставки алатке за структуре. Идите на језичак **Управљање структурама** и обезбедите да је одговарајућа дефиниција структуре активирана. Затим пређите на језичак **Структуре** и проверите да ли је жељени елемент приказан на десној страни.

## 3.2.3 Делење дефиниција структура

За уобичајене структуре можда не морате сами стварати дефиниције, већ се можете послужити постојећим дефиницијама са места попут [store.kde.org](http://store.kde.org).

Такође можете пожелети да поделите своје дефиниције. То чините тако што направите архиву (нпр. типа `.tar.gz`) која садржи само потфасциклу са `.desktop` фајлом и фајлом дефиниције структуре. Настављајући пример из претходног одељка, ово би била фасцикла `PrimerStruktur` са својим потпуним садржајем. Употреба овог формата за делење дефиниција структура омогућава њихово инсталирање у Октеу без ручних захвата.

## 3.2.4 Стварање дефиниција структура

### ПРИМЕДБА

Новији, али још недовршени, водич за писање дефиниција структура може се наћи на [викију Корисничке базе KDE-а](#).

Дефиниције структура могу се стварати на два начина. Први је писање дефиниције у ИксМЛ-у, а други употребом јаваскрипта. Приступ јаваскриптом омогућава вам да израдите сложеније структуре, са могућностима попут, рецимо, оверавања структуре. ИксМЛ пружа мање могућности, али може бити најлакши приступ ако је статичка структура све што

вам треба. Ако желите динамичку структуру, нпр. где дужине низова зависе од других вредности из структуре, или се распоред структуре мења са променом вредности неких чланова, мораћете да напишете дефиницију структуре на јаваскрипту. Овоме има један изузетак: ако дужина низа треба да буде **тачно** једнака некој другој вредности из структуре, то се може задати и ИксМЛ-ом; али ако је нешто попут *дужина - 1*, тада мора јаваскрипт.

### 3.2.5 ИксМЛ формат фајла за дефинисање структура

#### ПРИМЕДБА

Новији, али још недовршени, водич за писање дефиниција структура може се наћи на [викију Корисничке базе КДЕ-а](#).

.osd ИксМЛ фајл има један корени елемент: `<data>`, без атрибута. Унутар овог елемента мора стојати један од следећих:

#### `<primitive>`

За стварање примитивних типова података, попут *int* или *float*. Овај елемент нема поделемената, а може имати следеће атрибуте:

#### **type**

Тип овог примитивног типа. Мора бити један од следећих:

- *char* за 8-битни аски знак
- *int8*, *int16*, *int32*, *int64* за означени целобројни те величине
- *uint8*, *uint16*, *uint32*, *uint64* за неозначени целобројни те величине
- *bool8*, *bool16*, *bool32*, *bool64* за неозначени логички (0 = нетачно, било шта друго = тачно) те величине
- *float* за 32-битни број у покретном зарезу (ИЕЕЕ 754)
- *double* за 64-битни број у покретном зарезу (ИЕЕЕ 754)

#### `<bitfield>`

За стварање битског поља. Овај елемент нема поделемената, а може имати следеће атрибуте:

#### **width**

Број битова у овом битском пољу. Мора бити између 1 и 64.

#### **type**

Тип овог битског поља. Мора бити један од следећих:

- *unsigned* за битско поље где вредност треба тумачити као неозначену (опсег 0 до  $2^{\text{width}} - 1$ )
- *signed* за битско поље где вредност треба тумачити као означену (опсег  $-2^{\text{width}} - 1$  до  $2^{\text{width}} - 1$ )
- *bool* за битско поље где вредност треба тумачити као логичку

#### ПРИМЕДБА

Не заборавите да додате испуну после `<bitfield>`, пошто ће у супротном наредни елемент почети усред бајта (осим код ниски и низова, који аутоматски додају испуну). Наравно, испуна није потребна ако желите ово понашање.

#### `<enum>`

За стварање примитивног типа, али где су вредности приказане као чланови набрајања, ако је могуће. Овај елемент нема поделемената (али ће вам требати ознака `<enumDef>` у фајлу раду упућивања). Има следеће атрибуте:

**enum**

Подложно набрајање за ову вредност. Мора одговарати атрибуту *name* једне од ознака `<enumDef>` у овом фајлу.

**type**

Тип овог набрајања. Погледајте истоимени атрибут уз `<primitive>`. Једина разлика је то што овде *double* и *float* немају смисла.

`<flags>`

Исто као и `<enum>`, само што се вредности представљају као резултат *битског или* над свим вредностима набрајања.

`<struct>`

За стварање структуре. Сви други елементи (укључујући и `<struct>`) могу бити деца овога, и тада ће бити део резултујуће структуре.

`<union>`

За стварање уније. У суштини исто што и `<struct>`, осим што ће сви дечји елементи почињати од истог помака. Корисно за тумачење истог низа бајтова на различите начине.

`<array>`

За стварање низа. Овај елемент прихвата тачно једно дете (подложни тип низа), које може бити било који од елемената, чак и сам `<array>`. Има следеће атрибуте:

**length**

Број елемената у овом низу, као декадни број. Алтернативно може бити ниска једнака атрибуту *name* неког од претходно дефинисаних елемената `<primitive>`, `<enum>` или `<flags>`; тада ће дужина бити вредност тог елемента. Дужина је тренутно ограничена на 10000, јер би већи низови трошили превише меморије и превише успоравали програм.

`<string>`

За стварање ниске са одређеним кодирањем. Подразумевано се добија ниска окончана нулом, у стилу Ц-а. За стварање различитих типова ниски могу се употребити следећи атрибути:

**terminatedBy**

Овај атрибут одређује којом ће се уникодском кодном тачком ниска окончавати. Мора бити хексадекадни број (опционо са водећим *0x*); ако је кодирање аски, смислене су само вредности до *0x7f*. Ако није задато ни ово ни *maxCharCount* ни *maxByteCount*, претпоставља се 0 (ниска Ц стила).

**maxCharCount**

Највећи број знакова који ниска може да има. Ако је задато и *terminatedBy*, прво што наиђе окончава ниску. Међусобно искључиво са *maxByteCount*.

**maxByteCount**

Колико највише бајтова ниска може бити дугачка. Ако је задато и *terminatedBy*, прво што наиђе окончава ниску. Међусобно искључиво са *maxCharCount*. За кодирања попут аскија, ово је исто као *maxCharCount*.

**type**

Кодирање ове ниске. Може бити једно од следећих:

- *ASCII*
- *LATIN-1*
- *UTF-8*
- *UTF-16-LE* или *UTF-16-BE*. Ако се не зада ни суфикс *-LE* ни *-BE*, претпоставља се мала крајност.
- *UTF-32-LE* или *UTF-32-BE*. Ако се не зада ни суфикс *-LE* ни *-BE*, претпоставља се мала крајност.

Сваки елемент такође прихвата атрибут *name*, који се после види у приказу структура.

## 3.2.6 Пример исте структуре у ИксМЛ-у и јаваскрипту

### ПРИМЕДБА

Новији, али још недовршени, водич за писање дефиниција структура може се наћи на [викију Корисничке базе KDE-а](#).

### 3.2.6.1 Заједнички кораци за оба приступа

Наш фајл са метаподацима изгледа овако:

```
[Desktop Entry]
Encoding=UTF-8
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=Pera Peric
X-KDE-PluginInfo-Email=pera.peric@nikadodjija.org
X-KDE-PluginInfo-Name=simplestruct
X-KDE-PluginInfo-Version=1.0
X-KDE-PluginInfo-Website=http://www.nikadodjija.org/plugins/
X-KDE-PluginInfo-Category=structure
X-KDE-PluginInfo-License=LGPL
X-KDE-PluginInfo-EnabledByDefault=false
```

- ❶ Иконица коју Октеа приказује за ову структуру може бити нека коју проналази `kdialog --geticon`, или путања до иконице.

Сва поља би требало да су прилично очигледна, осим `X-KDE-PluginInfo-Name`. Вредност овог поља мора бити једнака имену фасцикле која садржи фајл, као и имену `.desktop` фајла. Ако се прави ИксМЛ дефиниција структуре, и име `.osd` фајла мора бити једнако овоме.

У овом примеру имали бисмо фасциклу по имену `simplestruct`, која садржи фајл `simplestruct.desktop`. Ако се структура дефинише ИксМЛ-ом, фасцикла би такође садржала фајл `simplestruct.osd`. Ако бисмо користили јаваскрипт, фајл дефиниције би се уместо тога звао `main.js`.

### 3.2.6.2 Једноставна ИксМЛ дефиниција структуре

За почетак дефинишимо врло једноставну пробну структуру која садржи само интегралне типове података (један знак, један 32-битни означени цео број и једно битско поље). У Ц-у или Ц++у ово би се изразило као:

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

Први корак је писање `.osd` фајла, формата дефинисаног у претходном одељку. Назваћемо га `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>
```

Ово је сасвим слично дефиницији у Ц-у и Ц++у

Направите фасциклу `simplestruct` унутар инсталационе фасцикле структура (погледајте одељак о ручном инсталирању дефиниција) и копирајте у њу ова два фајла. Сада можете поново покренути Октеу и користити нову структуру.

### 3.2.6.3 Једноставна структура у јаваскрипту

Да бисмо извели исту структуру помоћу јаваскрипта, направимо фајл по имену `main.js` (уместо претходног `simplestruct.osd`) и променимо `X-KDE-PluginInfo-Category=structure` у `X-KDE-PluginInfo-Category=structure/js`. Фајл треба да има следећи садржај:

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

Структура коју приказује Октеа увек је повратна вредност функције `init`.

Следеће функције се могу позивати ради стварања примитивних типова:

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

Функција `bitfield` узима два аргумента. Први је ниска `"bool"`, `"signed"` или `"unsigned"`. Други аргумент је цео број којим се задаје ширина у битовима.

### 3.2.6.4 Сложеније структуре

Прелазимо на дефинисање сложеније структуре, коју ћемо назвати `complex` и сачувати у фајл по имену `complex.osd`. Ова структура ће садржати два низа (један фиксне дужине а други коме се дужина одређује при извршавању), као и угнеђену структуру и унију.

```
<?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>
```

Ово би одговарало следећем у псеудо-Ц-у/Ц++-у:

```
struct complex {
    uint8_t size;
    union aUnion {
        int8_t fourBytes[4];
    };
    struct nested {
        char string[size]; // није правилан Ц++, упућује на вредно ←
        uint8_t size;
    };
};
```

#### ПРИМЕДБА

Очигледно само низови динамичке дужине могу упућивати на поља пре низа.

Затим правимо фајл `complex.desktop` као у претходном примеру (не заборавите да исправно задате `X-KDE-PluginInfo-Name`) и на исти начин инсталирамо оба фајла.

### 3.2.6.5 Још информација

Неколико примера дефиниција структура може се наћи [гит ризници](#). Међу њима су, на пример, дефиниције заглавље ПНГ и ЕЈФ фајлова. ИксМЛ шема која описује структуру `.osd` фајла такође се налази у [ризници](#). За још информација можете писати на [arichardson.kde@gmail.com](mailto:arichardson.kde@gmail.com).

## Глава 4

# Преглед сучеља

### 4.1 Ставке менија

Поред уобичајених КДЕ менија описаних у поглављу о менијима Основа КДЕ-а, Оклетта садржи и следеће програмски посебне ставке:

#### 4.1.1 Мени Фајл

##### Фајл → Нови (Ctrl+N)

Ствара нови низ бајтова...

- **празан:** ...без података.
- **из клипборда:** ...према текућем садржају клипборда.
- **образац...:** ...по датом обрасцу.
- **насумични подаци...:** ...са насумичним подацима.
- **низ:** ...са свим бајтовима од 0 до 255.

##### Фајл → Извоз

Извози изабране бајтове у фајл...

- **вредности:** ...кодирание као вредности бајтова. Подразумевано су вредности раздвојене једним размаком. Знак за раздвајање може да се промени кроз дијалог **Извези**.
- **знакови:** ...кодирание као обични текст.
- **база64:** ...кодирание у формату база64.
- **база32:** ...кодирание у формату база32.
- **аски-85:** ... кодирание у формату аски-85.
- **уу-кодирание:** ...кодирание у формату уу-кодирание.
- **иксикс-кодирание:** ...кодирание у формату иксикс-кодирание.
- **Интелов ХЕКС:** ...кодирание у формату Интеловог ХЕКС-а.
- **с-рекорд:** ...кодирание у формату с-рекорд.
- **Ц низ:** ...дефинисане као низ у програмском језику Ц.
- **обични текст (приказ):** ...као у приказу података, са помаком, вредностима бајтова и знаковима.

##### Фајл → Дозволе → Само-за-читање

Кад је постављено, учитани низ бајтова не може да се мења.

##### Фајл → Затвори све друге

Затвара све осим текућег низа бајтова.

## 4.1.2 Мени Уређивање

### Уређивање → Копирање као

Копира изабране бајтове у клипборд, у задатом формату. Списак доступних формата даје ставка менија **Фајл** → **Извези**.

### Уређивање → Уметни

#### Уметни образац...

Умеће задати низ бајтова код курсора.

Дијалог садржи опције за задавање броја уметања и формата обрасца (хексадекадно, декадно, октално, бинарно, знакови, УТФ-8).

### Уређивање → Поништи избор (Ctrl+Shift+A)

Поништава тренутни избор.

### Уређивање → Изабери опсег... (Ctrl+E)

Отвара угнежђени дијалог за уношење опсега за избор.

### Уређивање → Режим пребрисавања (Insert)

Пребацује између режима уметања и пребрисавања.

#### ПРИМЕДБА

Режим пребрисавања изведен је врло строго. Није могућа измена величине података (бајтови се немогу надовезивати нити уклањати).

### Уређивање → Нађи... (Ctrl+F)

Тражи задати образац кроз документ.. Могу се тражити хексадекадни, октални, бинарни и текстуални обрасци.

Кроз опције у дијалогу можете задати почетну тачку, смер и опсег претраге.

### Уређивање → На помак... (Ctrl+G)

Помера курсор на задати помак.

## 4.1.3 Мени Приказ

### Приказ → Приказ помака редова (F11)

Укључује и искључује приказ помака редова лево од окна.

### Приказ → Приказ вредности или знакова

Одређује које је од могућих тумачења бајтова приказано, једно од:

- Вредности
- Знакови
- Вредности и знакови

### Приказ → Кодирање вредности

Одређује кодирање вредности, једно од:

- хексадекадно
- декадно
- октално
- бинарно



#### Приказ → Кодирање знакова

Одређује кодирање знакова, једно од датих у подменију.

#### Приказ → Без неисписивих знакова

Укључује и искључује приказ невидљивих знакова.. Ако су укључени, на одговарајућим местима у колони знакова приказан је заменски знак.

#### Приказ → Постави бајтове по реду...

Изаберите колико се бајтова приказује по реду, подразумевано је то 16.

#### Приказ → Постави бајтове по групи...

Подразумевано се хексадекадне вредности приказују у групама од по 4 бајта. Овом ставком менија можете то променити.

#### Приказ → Динамички распоред

Задаје правила за распоред приказа података. Овим се дефинише број бајтова по једном реду у зависности од ширине приказа. Могућа правила су:

- **Искључен:** распоред је фиксиран на текући број бајтова по реду и не прилагођава се изменама величине приказа.
- **Преламај само целе групе бајтова:** ставља највећи могући број бајтова у један ред, али тако да одржи групе бајтова потпуним.
- **Укључен:** исто као претходно, али допушта непотпуне групе бајтова.

#### Приказ → Режим приказа

Одређује распоред приказа, један од:

- **Колоне:** вредности и тумачења знакова приказани су у класичном распореду, свако у својој засебној колони.
- **Врсте:** знаковно тумачење бајта приказано је непосредно испод вредносног тумачења.

#### Приказ → Подели водоравно (Ctrl+Shift+T)

Дели област тренутно фокусираног приказа на два дела и додаје копију текућег приказа у нову, нижу област.

#### Приказ → Подели усправно (Ctrl+Shift+L)

Дели област тренутно фокусираног приказа на два дела и додаје копију текућег приказа у нову, десну област.

#### Приказ → Затвори област приказа (Ctrl+Shift+R)

Затвара област тренутно фокусираног приказа.

#### Приказ → Режим приказа

Поставке приказа могу да се засебно ускладиште као профил приказа. Тренутно изабрани профил може да се ажурира непосредно у поставкама текућег приказа, или се на основу њих може направити нови профил. Профилима приказа управља се у дијалогу на **Поставке → Управљај профилима приказа...**

### 4.1.4 Мени Прозори

Пружа списак тренутних приказа. Одавде бирате активан прозор.

#### 4.1.5 Мени Обележивачи

На исти низ бајтова може се поставити више обележивача. Сваки низ бајтова има свој скуп обележивача, који је приказан на дну менија **Обележивачи**. Избором обележивача из овог менија померате курсор и приказ за њега.

**ПРИМЕДБА**

Обележивачи су за сада пролазни, тј. не уписују се када затворите низ бајтова или цео програм.

**Обележивачи** → **Додај обележивач (Ctrl+B)**

Обележи локацију унутар низа бајтова.

**Обележивачи** → **Уклони обележивач (Ctrl+Shift+B)**

Уклања текући обележивач. Ова наредба је расположива само ако је курсор на обележеној локацији.

**Обележивачи** → **Уклони све обележиваче**

Чисти списак обележивача.

**Обележивачи** → **На претходни обележивач (Alt+Up)**

Помера курсор на претходни обележивач.

**Обележивачи** → **На следећи обележивач (Alt+Down)**

Помера курсор на следећи обележивач.

#### 4.1.6 Мени Алатке

Списак инсталираних алатки, кроз који се може укључити и искључити приказ сваке од њих. Детаљан опис сваке алатке налази се у секцији [о алаткама](#).

#### 4.1.7 Мени Поставке

**Поставке** → **Управљај профилима приказа...**

Отвара дијалог за стварање, уређивање и брисање профила приказа, као и постављање подразумеваног.

## Глава 5

# Заслуге и лиценца

Октета

(програм) © 2006–2012, Фридрих В.Х. Косебау [kossebau@kde.org](mailto:kossebau@kde.org).

(документација) © 2008, 2010, Фридрих В.Х. Косебау [kossebau@kde.org](mailto:kossebau@kde.org), Александер Ричардсон [arichardson.kde@gmail.com](mailto:arichardson.kde@gmail.com).

Прево Драган Пантелић [falcon-10@gmx.de](mailto:falcon-10@gmx.de).

Документација се даје на коришћење под условима [Гнуове Лиценце слободне документације](#).

Програм се даје на коришћење под условима [Гнуове Опште јавне лиценце](#).