Manuale di Okteta

Friedrich W. H. Kossebau
Alex Richardson
Traduzione della documentazione: Federico Zenith

Indice

Sommario

Okteta è un semplice editor dei dati grezzi di un file. Questo tipo di programma si chiama anche editor esadecimale o binario.

Capitolo 1

Introduzione

Okteta è un semplice editor dei dati grezzi di un file.

I dati sono visualizzati in due modi: come i valori numerici dei byte e come i caratteri loro assegnati. I valori e i caratteri possono essere visualizzati in due colonne (la visualizzazione normale degli editor esadecimali) o in righe con il valore sopra il carattere. Le modifiche possono essere effettuate sia sui valori che sui caratteri.

Oltre alle solite funzionalità di modifica, Okteta include una piccola serie di strumenti, come una tabella che elenca le decodifiche in tipi di dati semplici e comuni, una che elenca tutti i byte possibili con i loro equivalenti in caratteri e valori, una vista informativa con statistiche, un calcolatore di codici di controllo, strumenti per il filtraggio e l'estrazione di stringhe.

Tutte le modifiche ai dati caricati possono essere annullate o rifatte senza limiti.

Capitolo 2

Fondamentali

2.1 Avviare Okteta

Scrivi **`okteta`** sulla riga di comando, o seleziona **Editor esadecimale** dal gruppo **Applicazioni** → **Accessori** nell'avviatore di applicazioni.

Sono disponibili le opzioni da riga di comando standard di Qt™ e KDE Frameworks 5, che possono essere elencate scrivendo **`okteta --help`**.

Le opzioni da riga di comando specifiche a Okteta sono:

`<URL>`: apri file dallo o dagli URL specificati

2.2 Uso

La finestra principale di Okteta ha i seguenti componenti: una barra dei menu, una barra degli strumenti, barra di stato, una o più barre laterali con strumenti, e la vista principale con le viste dei dati organizzate per schede.

Quando si apre un file o si crea un nuovo vettore di byte, i byte ivi contenuti vengono elencati di fila in righe con un certo numero di byte per riga. Sono visualizzati in due modi: come i valori numerici dei byte e come i caratteri loro assegnati. I valori e i caratteri possono essere visualizzati separati in due colonne o fianco a fianco con il valore sopra il carattere. Sul lato sinistro sono indicati gli scostamenti del primo byte di ogni riga.

La gestione è simile alla maggior parte degli editor di testo: i dati possono essere modificati, tagliati, copiati, incollati, trascinati più o meno come il testo in questi. Un cursore indica la posizione attuale. Premere il pulsante **Ins** fa passare tra le modalità di inserimento e sovrascrittura. La modalità di sovrascrittura è più rigorosa che negli editor di testo, e non permette nessuna operazione che cambi le dimensioni del vettore di byte.

Diversamente dagli editor di testo, i contenuti sono visualizzati in due modi. Per l'inserimento si può usare solo uno di questi. Ci sono due cursori collegati, mostrati per la visualizzazione dei valori e dei caratteri; il cursore della visualizzazione attiva lampeggia. Quando sono attivi i caratteri, li si può inserire come in un editor di testo. Quando sono attivi i valori, scrivere una numero farà aprire un editor minimo per inserire il resto del valore.

La finestra di ricerca permette all'utente di cercare una stringa specifica di byte, definibile come valori (esadecimale, decimale, ottale, binaria) o testo (codifica a 8 bit corrente o UTF-8).

Si possono aprire contemporaneamente più vettori di dati, ma solo uno può essere attivo. Usa il menu **Finestre** per selezionare quale vettore debba essere attivo.

Capitolo 3

Strumenti

3.1 Panoramica

Okteta include alcuni strumenti, alcuni per analizzare e manipolare i vettori di dati e altri con scopi più generali. Questi strumenti si possono attivare o disattivare dal menu **Strumenti**. Ogni strumento ha una piccola vista, che si aggancia in una delle barre laterali o si presenta come finestra a sé. Puoi agganciare, sganciare, ridisporre e anche impilare le viste degli strumenti col mouse, premendo il tasto sinistro del mouse sulla barra del titolo di una vista, spostandola dove vuoi e rilasciando il tasto sinistro del mouse per completare l'azione, altrimenti annullala premendo il tasto **Esc**.

3.1.1 Analisi e manipolazione

Tabella di valori e caratteri

La tabella elenca tutti i valori possibili dei byte, sia come caratteri che con i diversi codici numerici.

Il valore selezionato può essere inserito alla posizione del cursore per un certo numero di byte. Questo si può fare usando il pulsante **Ins** o facendo doppio clic sulla riga nella tabella.

Filtro binario

Il filtro effettua operazioni binarie sui byte selezionati. Dopo aver scelto l'operazione (ET, VEL, RUOTA...), gli eventuali parametri possono essere impostati nella casella sotto. Il filtro viene eseguito all'attivazione del pulsante **Filtra**.

Stringhe

Questo strumento localizza le stringhe nei byte selezionati. Dopo aver scelto la lunghezza minima delle stringhe, queste vengono ricercate dopo aver usato il pulsante **Estrai**. L'elenco di stringhe visualizzate può venire ristretto inserendo un termine di filtro.

Statistiche

Questo strumento genera un dato statistico dei byte selezionati: la frequenza di ogni byte nella selezione. Può essere calcolata con il pulsante **Genera**.

Codice di controllo

Questo strumento calcola vari codici di controllo per i byte selezionati. Dopo aver scelto l'algoritmo e impostato il parametro, se ce n'è uno, la somma viene calcolata alla pressione del pulsante **Calcola**.

Tabella di decodifica
La tabella visualizza i valori dei byte a partire dal cursore per alcuni tipi di dati comuni come numeri interi o a virgola mobile, ma anche UTF-8. Fare doppio clic in una riga della tabella apre un editor per modificare il valore.

Strutture
Questo strumento permette di investigare e modificare vettori di dati basati su definizioni di strutture dell'utente. Per istruzioni dettagliate vedi Sezione 3.2.

3.1.2 Strumenti generali

Filesystem
Questo strumento offre un selettore di file incorporato.

Documenti
Questo strumento mostra tutti i file attualmente creati o caricati. I simboli indicano il file la cui vista è attiva e anche quali file hanno modifiche non salvate o la cui copia salvata è stata modificata da un altro programma.

Segnalibri
Questo strumento può essere usato per gestire i segnalibri, in alternativa al menu **Segnalibri**.

> NOTA
> I segnalibri sono al momento temporanei e non vengono salvati se chiudi un vettore di byte o tutto il programma.

Informazioni sul file
Questo strumento visualizza alcune informazioni sul file attuale, inclusi il tipo, il luogo di memorizzazione e le dimensioni.

Terminale
Un terminale incorporato; la cartella di lavoro non è collegata al file attivo.

Conversione di insieme di caratteri
Lo strumento riscrive i byte in modo che i rispettivi caratteri siano gli stessi dell'altro insieme di caratteri. Sono supportati solo insiemi di caratteri di 8 bit, e i caratteri che non corrispondono vengono attualmente sostituiti da un valore fisso a 0.

3.2 Strumento per le strutture

3.2.1 Generale

Lo strumento per le strutture permette di analizzare e modificare i vettori di byte in base a definizioni di strutture dell'utente, che possono essere costituite da array, unioni, tipi primitivi ed enumerazioni.
Ha una propria finestra di configurazione, che si può raggiungere con il pulsante **Impostazioni**. Ci sono diverse opzioni configurabili, come lo stile con cui i valori sono visualizzati (decimale, esadecimale, o binario). Inoltre è possibile scegliere quali definizioni di strutture caricare e quali mostrare nella vista.
Le strutture sono definite in file di definizione di strutture di Okteta (basati su XML, con estensione `.osd`). Inoltre, un file `.desktop` contiene metadati sul file di descrizione delle strutture, come autore, pagina Web e licenza.

Attualmente non c'è un supporto integrato per creare o modificare definizioni di strutture, quindi ciò va fatto manualmente, come descritto nelle sezioni seguenti.

3.2.2 Installare definizioni di strutture

3.2.2.1 Installare con le Novità

Il modo più facile di installare nuove definizioni di strutture è usare il supporto per le Novità incorporato in Okteta. Per installare una struttura esistente apri la finestra delle impostazioni dello strumento per le strutture. Lì, seleziona la scheda **Gestione strutture** e premi il pulsante **Prendi nuove strutture**. La finestra che apparirà permette di installare e disinstallare strutture.

3.2.2.2 Installare manualmente definizioni di strutture

Lo strumento per le strutture cerca descrizioni di strutture nella sotto-cartella `okteta/structures/` della cartella dei dati dei programmi dell'utente (si trova con **`qtpaths --paths GenericDataLocation`**). Potresti dover creare questa cartella se non ci sono ancora definizioni di strutture installate.

Ci sono due file per ogni definizione di struttura: un file per la definizione vera e propria, e un file `.desktop` per i metadati (autore, versione, eccetera).

In quella cartella c'è una sottocartella per ogni definizione di struttura, contenente sia il file `.desktop` che il file `.osd` o `main.js` della definizione.

Per esempio, con la cartella di dati dei programmi `qtpaths --paths GenericDataLocation` e una definizione di struttura chiamata Esempio, c'è la cartella `okteta/structures/Esempio`, contenente un file `Esempio.desktop` e un file `Esempio.osd`.

3.2.2.3 Usare le strutture appena installate

Se hai installato una nuova definizione di struttura creando una tale sottocartella con i due file o li hai modificati, devi riavviare Okteta e aprire la finestra di configurazione dello strumento per le strutture. Lì, seleziona la scheda **Gestione strutture** e assicurati che la definizione di struttura desiderata sia segnata. Usa quindi il pulsante **Applica**, passa alla scheda **Strutture** e assicurati che l'elemento desiderato sia elencato sul lato destro.

3.2.3 Condividere definizioni di strutture

Per le strutture comuni potresti non dover creare una definizione personalizzata, ma riutilizzarne invece una già disponibile per esempio da store.kde.org.

Potresti anche voler condividere una tua definizione. Per farlo, crea un file d'archivio (per esempio un archivio tar compresso, `.tar.gz`) contenente solo la sottocartella con il file `.desktop` e il file di definizione della struttura. Nell'esempio nella sezione precedente sarebbe nella cartella `Esempio` con tutti i suoi contenuti. Usare questo formato per condividere le definizioni delle strutture permette di installarle da dentro Okteta e non richiede l'installazione manuale.

3.2.4 Creare definizioni di strutture

> NOTA
> Una guida più aggiornata, ma non completa, per la scrittura di definizioni di strutture può essere consultata sul wiki KDE UserBase.

Ci sono due modi diversi di creare definizioni di strutture. La prima è scrivere la definizione in XML, l'altra è usare JavaScript. L'approccio con JavaScript permette di creare strutture più complesse con funzionalità come per esempio la validazione della struttura. Usare XML dà meno

opportunità, ma se ti serve solo una struttura statica potrebbe essere la cosa più facile. Se ti serve una struttura dinamica, per esempio in cui le lunghezze dei vettori dipendono da altri valori nella struttura, o la disposizione della struttura è diversa quando qualche valore membro cambia, dovrai scrivere la definizione della struttura in JavaScript. C'è un'eccezione a questa regola: se hai un array in cui la lunghezza deve essere **esattamente** la stessa di un altro valore nella struttura, puoi usare XML. Se invece è qualcosa come **`lunghezza - 1`** devi ancora usare JavaScript.

3.2.5 Formato dei file XML per le definizioni di strutture

> NOTA
> Una guida più aggiornata, ma non completa, per la scrittura di definizioni di strutture può essere consultata sul wiki KDE UserBase.

Il file XML `.osd` ha un elemento radice: **`<data>`**, senza attributi. Dentro a questo elemento ci dev'essere uno dei seguenti elementi:

<primitive>

Per creare un tipo di dati primitivo, come per esempio **`int`** e **`float`**. Questo elemento non accetta sottoelementi e può avere i seguenti attributi:

type

Il tipo di questo tipo primitivo. Deve essere uno dei seguenti:

- **`char`** per un carattere ASCII a 8 bit
- **`int8, int16, int32, int64`** per un intero con segno di quella dimensione
- **`uint8, uint16, uint32, uint64`** per un intero senza segno di quella dimensione
- **`bool8, bool16, bool32, bool64`** per un booleano senza segno (0 = falso, qualsiasi altro valore = vero) di quella dimensione
- **`float`** per un numero a virgola mobile IEEE754 a 32 bit
- **`double`** per un numero a virgola mobile IEEE754 a 64 bit

<bitfield>

Per creare un campo di bit. Questo elemento non accetta sottoelementi e può avere i seguenti attributi:

width

Il numero di bit usati da questo campo. Deve essere tra 1 e 64.

type

Il tipo di questo campo di bit. Deve essere uno dei seguenti:

- **`unsigned`** per un campo di bit dove il valore verrà interpretato come senza segno (valore tra 0 e $2^{\text{larghezza}}$ - 1)
- **`signed`** per un campo di bit dove il valore verrà interpretato come con segno (valore tra - $2^{\text{larghezza - 1}}$ e $2^{\text{larghezza - 1}}$ - 1)
- **`bool`** per un campo di bit dove il valore verrà interpretato come un booleano

> NOTA
> Ricordati sempre di aggiungere dello spazio dopo un **`<bitfield>`**, perché altrimenti il prossimo elemento (tranne che per stringhe ed array, visto che questi aggiungono spazio automaticamente) comincerà nel mezzo di un byte. Ovviamente lo spazio non è necessario se desideri questo comportamento.

<enum>

Per creare un tipo primitivo, ma in cui i valori siano visualizzati come membri di un'enumerazione, se possibile. Questo elemento non accetta nessun sottoelemento (servirà però un'etichetta **`<enumDef>`** nel file per farvi riferimento). Ha i seguenti attributi:

enum

L'enumerazione su cui si basa questo valore. Deve corrispondere all'attributo **`name`** delle etichette **`<enumDef>`** in questo file.

type

Il tipo dell'enumerazione. Vedi l'attributo omonimo di **`<primitive>`**. L'unica differenza è che **`Double`** e **`Float`** non hanno senso.

<flags>

È la stessa cosa che **`<enum>`** con la sola differenza che i valori sono rappresentati come un *O logico sui bit* di tutti i valori dell'enumerazione.

<struct>

Per creare una struttura. Tutti gli altri elementi (incluso **`<struct>`**) possono esserne figli, e saranno parte della struttura risultante.

<union>

Per creare un'unione. Sostanzialmente la stessa cosa di **`<struct>`**, tranne che tutti gli elementi figli partiranno dallo stesso scostamento. È utile per interpretare la stessa sequenza di byte in modi diversi.

<array>

Per creare un array. Questo elemento accetta esattamente un figlio (il tipo di array su cui si basa), che può essere qualsiasi elemento, anche **`<array>`** stesso. Ha anche i seguenti attributi:

length

Il numero di elementi in questo array come numero decimale. In alternativa può anche essere una stringa corrispondente al nome di un **`<primitive>`**, **`<enum>`** o **`<flags>`** definito in precedenza. La lunghezza sarà in tal caso sempre uguale al valore di quell'elemento. Attualmente è limitata a 10000, perché array di maggiori dimensioni userebbero troppa memoria e rallenterebbero troppo lo strumento.

<string>

Per creare una stringa in varie codifiche. Come impostazione predefinita ottieni una stringa terminata da *NULL* in stile C. È però possibile creare diversi tipi di stringa con i seguenti attributi:

terminatedBy

Questo attributo determina quale punto di codice Unicode conclude la stringa. Deve essere un numero esadecimale (facoltativamente con un **`0x`** iniziale). Quando la codifica è ASCII, solo i valori fino a **`0x7f`** hanno senso. Se né questo valore, né **`maxCharCount`** né **`maxByteCount`** sono impostati, si assume che sia 0 (stringa in stile C).

maxCharCount

Il numero massimo di caratteri di questa stringa. Se è impostato anche **`terminatedBy`**, il primo criterio a verificarsi termina la stringa. Questo è mutuamente esclusivo con **`maxByteCount`**.

maxByteCount

La lunghezza massima in byte di questa stringa. Se è impostato anche **`terminatedBy`**, il primo criterio a verificarsi termina la stringa. Questo è mutuamente esclusivo con **`maxCharCount`**. Con codifiche come **`ASCII`** è la stessa cosa di **`maxCharCount`**.

type

La codifica di questa stringa. Può essere una delle seguenti:

- *ASCII*
- *LATIN-1*
- *UTF-8*
- **`UTF-16-LE`** o **`UTF-16-BE`**. Se non viene dato né il suffisso **`-LE`** né **`-BE`**, si assume la codifica *little endian*.
- **`UTF-32-LE`** o **`UTF-32-BE`**. Se non viene dato né il suffisso **`-LE`** né **`-BE`**, si assume la codifica *little endian*.

Ogni elemento accetta anche un attributo **`name`** che è poi visibile nella vista delle strutture.

3.2.6 Una definizione di struttura d'esempio in XML e in JavaScript

> NOTA
> Una guida più aggiornata, ma non completa, per la scrittura di definizioni di strutture può essere consultata sul wiki KDE UserBase.

3.2.6.1 Il passo in comune ad entrambi gli approcci

Il nostro file di metadati ha questo aspetto:

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

Name=Semplice struttura di prova
Comment=Struttura di prova molto semplice con due soli elementi

X-KDE-PluginInfo-Author=Pinco Pallino
X-KDE-PluginInfo-Email=pinco@pallino.it
X-KDE-PluginInfo-Name=semplice
X-KDE-PluginInfo-Version=1.0
X-KDE-PluginInfo-Website=https://it.wiktionary.org/wiki/semplice
X-KDE-PluginInfo-Category=structure
X-KDE-PluginInfo-License=LGPL
X-KDE-PluginInfo-EnabledByDefault=false
```

❶ L'icona visualizzata in Okteta per questa struttura. Può essere qualsiasi cosa che si può trovare eseguendo **`kdialog --geticon`**, o il percorso a un'icona.

Questi campi dovrebbero essere tutti abbastanza facili da capire, tranne `X-KDE-PluginInfo-Name`: questo deve avere il nome della cartella contenente il file così come il nome del file `.desktop`. Quando si creano definizioni di strutture con XML, anche il nome del file `.osd` deve corrispondere.

In questo esempio avremmo una cartella di nome `semplice` contenente il file `semplice.desktop`. Quando si definiscono strutture in XML, la cartella conterrebbe anche un file di nome `semplice.osd`. Usando JavaScript, avremmo invece un file di nome `main.js`.

3.2.6.2 Una semplice definizione di struttura XML

Per cominciare creiamo una definizione per una struttura di prova molto semplice, contenente solo tipo di dati integrali (un carattere, un intero a 32 bit, e un campo di bit). In C e C++ verrebbero espressi come:

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

Il primo passo è scrivere il file `.osd` secondo il formato di file definito nella sezione precedente. Lo chiameremo `semplice.osd`:

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

il che è abbastanza simile alla definizione in C o C++.

Adesso crea una cartella `semplice` sotto la cartella di installazione delle strutture (rivedi come si installano manualmente le strutture), e copiaci i due file. Adesso puoi riavviare Okteta e usare la nuova struttura.

3.2.6.3 La struttura semplice in JavaScript

Per implementare la struttura sopra in JavaScript, crea un file di nome `main.js` invece che `s emplicestruttura.osd` e cambia X-KDE-PluginInfo-Category=structure in X-KDE-PluginInfo-Category=structure/js. I contenuti di questo file dovrebbero essere:

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

La struttura visualizzata da Okteta è sempre il valore restituito dalla funzione `init`.

Le funzioni seguenti possono essere chiamate per creare un tipo primitivo:

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

- `float()`
- `double()`

La funzione `bitfield` prende due argomenti, di cui il primo è una stringa che può essere `bool`, `signed` o `unsigned`. Il secondo è un intero che imposta l'ampiezza in bit.

3.2.6.4 Strutture più complesse

Adesso creiamo la definizione di una struttura più complessa, che chiameremo 'complessa' e salveremo in un file di nome `complessa.osd`. Questa struttura conterrà due array (uno di lunghezza fissa e uno di lunghezza determinata durante l'esecuzione), oltre a una struttura annidata e un'unione.

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
  <struct name="complessa">
    <primitive name="dimensione" type="UInt8" />
    <union name="unione">
      <array name="quattroByte" length="4">
        <primitive type="Int8" />
      </array>
    </union>
    <struct name="annidata">
      <array name="stringa" length="dimensione"> <!-- fa ↩
          riferimento alla dimensione di cui sopra -->
        <primitive type="Char" />
      </array>
    </struct>
  </struct>
</data>
```

Ciò corrisponderebbe a quanto segue in pseudo C o C++:

```
struct complessa {
        uint8_t dimensione;
        union unione {
          int8_t quattroByte[4];
        };
        struct annidata {
          char stringa[dimensione] // non è C++ valido, fa riferimento ↩
              al valore della dimensione uint8
        };
      };
```

> **NOTA**
> Ovviamente puoi far riferire array a lunghezza dinamica solo a campi che vengono prima di questi.

Adesso creiamo il file `complessa.desktop` come nell'esempio precedente (assicurati di impostare `X-KDE-PluginInfo-Name` correttamente) e facciamo lo stesso per installare i file.

3.2.6.5 Ulteriori informazioni

Alcune definizioni di strutture esemplificative sono reperibili nel deposito Git, incluse per esempio le intestazioni dei file PNG e per i file ELF. Uno schema XML che descrive la struttura del file `.osd` è disponibile qui. Se servono maggiori informazioni, non esitare a contattare arichardson.kde@gmail.com.

Capitolo 4

Panoramica dell'interfaccia

4.1 Voci dei menu

Oltre ai comuni menu di KDE descritti nel capitolo Menu dei Fondamenti di KDE, Okteta ha le seguenti voci specifiche:

4.1.1 Menu File

File → Nuovo (Ctrl+N)

Crea un nuovo vettore di byte

- **Vuoto**: ... come vettore vuoto.
- **Dagli appunti**: ... dai contenuti attuali degli appunti.
- **Schema**: ... con un certo schema.
- **Dati casuali**: ... con dati casuali.
- **Sequenza**: ... con tutti i byte da 0 a 255.

File → Esporta

Esporta i byte selezionati in un file:

- **Valori**: codificati come valori di byte. Come impostazione predefinita, i valori sono separati da uno spazio. I caratteri di **Separazione** si possono modificare nella finestra **Esporta**.
- **Caratteri**: codificati come testo semplice.
- **Base64**: codificati nel formato Base64.
- **Base32**: codificati nel formato Base32.
- **ASCII85**: codificati nel formato ASCII85.
- **Codifica Uuencode**: codificati nel formato Uuencode.
- **Codifica Xxencode**: codificati nel formato Xxencode.
- **Intel esadecimale**: codificati nel formato Intel HEX
- **S-record**: codificati nel formato S-record.
- **Array C**: definiti come un array nel linguaggio di programmazione C.
- **Vista in testo semplice**: come nella vista dati con slittamento, valori di byte e caratteri.

File → Permessi → Imposta a sola lettura

Se impostato, non si possono effettuare modifiche al vettore di byte caricato.

File → Chiudi tutti gli altri

Chiude tutti i vettori di byte tranne l'attuale.

4.1.2 Menu Modifica

Edit → Copia come

Copia i byte selezionati in uno dei diversi formati negli appunti. Per un elenco dei formati disponibili, vedi la voce del menu **File → Esporta**.

Modifica → Inserisci

Inserisci schema

Inserisci una stringa di byte specificata al cursore.

Le opzioni nella finestra permettono di specificare il numero di inserimento dello schema e il suo formato (esadecimale, decimale, ottale, binario, caratteri o UTF-8).

Modifica → Deseleziona (Ctrl+Shift+A)

Deseleziona la selezione attuale.

Modifica → Seleziona intervallo (Ctrl+E)

Apre una finestra incorporata per inserire l'intervallo da selezionare.

Modifica → Modalità di sovrascrittura (Ins)

Passa tra le modalità di inserimento e sovrascrittura.

> NOTA
> La modalità di sovrascrittura è implementata in modo da essere molto severa: non è possibile cambiare le dimensioni dei dati, quindi non se ne possono aggiungere né rimuovere.

Modifica → Trova (Ctrl+F)

Trova uno schema specifico nel documento. Si possono cercare schemi esadecimali, decimali, ottali, binari o di testo.

Le opzioni nella finestra di permettono di specificare il punto di partenza, la direzione e l'estensione della ricerca.

Modifica → Vai allo slittamento (Ctrl+G)

Sposta il cursore a un certo slittamento.

4.1.3 Menu Visualizza

Visualizza → Mostra slittamento delle linee (F11)

Attiva o disattiva la visualizzazione dello slittamento delle righe su un pannello a sinistra.

Visualizza → Mostra valori o caratteri

Seleziona quale interpretazione dei dati viene visualizzata. Le possibili sono:

- **Valori**
- **Caratteri**
- **Valori e caratteri**

Visualizza → Codifica dei valori

Seleziona la codifica dei valori tra:

- **Esadecimale**
- **Decimale**
- **Ottale**
- **Binario**

Visualizza → **Codifica dei caratteri**
Seleziona la codifica dei caratteri dal sottomenu.

Visualizza → **Mostra caratteri non stampabili**

Attiva o disattiva la visualizzazione dei caratteri non stampabili. Se la visualizzazione è disattivata, ai posti corrispondenti nella colonna dei caratteri verrà mostrato un carattere sostitutivo.

Visualizza → **Imposta byte per riga**

Seleziona i byte visualizzati per riga dalla finestra; il valore predefinito è 16 byte.

Visualizza → **Imposta byte per gruppo**

Da impostazione predefinita i valori esadecimali vengono visualizzati in gruppi di quattro byte. Con questo elemento del menu puoi modificare questo numero alle tue preferenze.

Visualizza → **Schema dinamico**
Imposta le regole per lo schema di visualizzazione dei dati. Questo definisce quanti byte visualizzare per riga, a seconda della larghezza della vista. Le regole possibili sono:

- **Disattivo**: lo schema è fisso all'attuale numero di byte per riga e non viene adattato ai cambiamenti di dimensioni della vista.
- **Raggruppa solo gruppi di byte completi**: mette il massimo possibile di byte per riga, fintanto che i gruppi di byte sono completi.
- **Attivo**: come il precedente, ma permette anche gruppi non completi di byte.

Visualizza → **Modalità di visualizzazione**
Seleziona lo schema di vista tra:

- **Colonne**: le interpretazioni a valori e a caratteri sono mostrate nello schema classico, con ciascuna in una colonna separata.
- **Righe**: l'interpretazione a caratteri di un byte è mostrata direttamente sotto a quella a valori.

Visualizza → **Dividi orizzontalmente (Ctrl+Shift+T)**
Divide l'area di visualizzazione della vista attiva in due parti e aggiungi una copia della vista attuale nella nuova area inferiore.

Visualizza → **Dividi verticalmente (Ctrl+Shift+L)**
Divide l'area di visualizzazione della vista attiva in due parti e aggiungi una copia della vista attuale nella nuova area a destra.

Visualizza → **Chiudi area della vista (Ctrl+Shift+R)**
Chiudi l'area di visualizzazione della vista attiva.

Visualizza → **Profilo di vista**
Le impostazioni di visualizzazione possono essere memorizzate a parte come profili di vista. Il profilo attualmente selezionato può essere aggiornato direttamente dalle impostazioni della vista attuale, o da queste se ne può creare una nuova. Tutti i profili di vista possono essere gestiti in una finestra disponibile da **Impostazioni** → **Gestisci i profili di vista**.

4.1.4 Menu Finestre

Fornisce un elenco delle viste attuali. Seleziona la finestra attiva.

4.1.5 Menu Segnalibri

Si possono impostare più segnalibri per ogni vettore di byte. Ogni vettore di byte ha la sua serie di segnalibri, e la serie appropriata viene visualizzata in fondo al menu **Segnalibri**. Scegline uno dal menu per spostarci il cursore e la vista.

> NOTA
> I segnalibri sono al momento temporanei e non vengono salvati se chiudi un vettore di byte o tutto il programma.

Segnalibri → **Aggiungi segnalibro (Ctrl+B)**

Segna una posizione nel vettore di dati.

Segnalibri → **Rimuovi segnalibro (Ctrl+Shift+B)**

Rimuovi il segnalibro attuale. Questo comando è disponibile solo se il cursore è a una posizione segnata.

Segnalibri → **Rimuovi tutti i segnalibri**

Pulisci l'elenco dei segnalibri.

Segnalibri → **Vai al segnalibro precedente (Alt+↑)**

Sposta il cursore al segnalibro precedente.

Segnalibri → **Vai al segnalibro successivo (Alt+↓)**

Sposta il cursore al prossimo segnalibro.

4.1.6 Menu Strumenti

Fornisce un elenco di strumenti installati. Attiva o disattiva la visualizzazione di ogni strumento. Trovi una descrizione dettagliata di ogni strumento nella sezione Strumenti.

4.1.7 Menu Impostazioni

Impostazioni → **Gestisci i profili di vista**

Apri una finestra per creare, modificare, eliminare ed impostare un profilo di vista predefinito.

Capitolo 5

Riconoscimenti e licenza

Okteta

Traduzione in italiano di Federico Zenith federico.zenith@member.fsf.org