MUP (autore: Vittorio Albertoni)

Premessa

Mup sta per Music publication ed è un software che ci consente di produrre partiture e spartiti musicali scrivendo istruzioni con la tastiera del computer in un particolare linguaggio.

Penso si tratti dell'antesignano dei software della specie.

Pare fosse già utilizzato nel 1992 da molti musicisti in tutto il mondo e nelle università e nel 1995 venne portato al grande pubblico, come shareware, dalla Arkkra Enterprises e i suoi sviluppatori decisero di renderlo open source nel 2012.

Purtroppo la Arkkra Enterprises, che fino a qualche tempo fa distribuiva installer del software per Linux, Windows e Mac, è scomparsa dal web e tutto ciò che resta disponibile è il source code su Github.

Fortunatamente possiamo installare ancora senza alcuna fatica questo software dal repository di Fedora Linux, che lo fornisce preinstallato su Fedora Jam. Con qualche accorgimento che vedremo nel manuale possiamo così procurarci il file .rpm che usa Fedora e installarlo su Debian e derivate (Ubuntu, MXLinux, ecc.).

In questo modo, almeno su Linux, abbiamo modo di utilizzare questo software, molto potente e capace di risultati qualitativamente vicinissimi alla perfezione del suo omologo Lilypond e superiori alla qualità dell'altro suo omologo ABC, entrambi comparsi solo due anni dopo, nel 1997 [1] .

Rispetto al software ABC, che già rappresenta un miracolo facendo le cose che fa impegnando solo pochissimi mega byte del nostro disco, Mup fa molte cose in più e meglio impegnando ancor meno mega byte.

[1] Per chi fosse interessato a queste alternative rimando ai seguenti manualetti che ho pubblicato sul mio blog all'indirizzo *www.vittal.it*.

Per quanto riguarda il linguaggio Lilypond il manualetto si intitola "lilypond" ed è allegato all'articolo del febbraio 2025 intitolato "Scrivere musica con LilyPond".

Per quanto riguarda il linguaggio ABC il manualetto si intitola "abc_notation" ed è allegato all'articolo del febbraio 2024 intitolato "Scrivere musica con ABC".

Indice

1 Installazione **3**

2 Come funziona **3**

3 Linguaggio MUP in sintesi **4**
3.1 Scrittura delle note 4
3.2 Scrittura del rigo musicale 5
3.3 Arricchimento del rigo 6
3.3.1 Testo del canto 6
3.3.2 Indicazioni dinamiche e varie 6
3.3.3 Indicazioni armoniche 7
3.3.4 Basso continuo 7
3.3.5 Indicazioni di analisi armonica 7
3.4 Spartiti e partiture 8
3.5 Riservato ai chitarristi 11
3.5.1 Intavolatura 11
3.5.2 Diagrammi degli accordi 12
3.6 Gestione del MIDI 13

4 Alcuni esempi **13**
4.1 Spartito per pianoforte con analisi armonica 13
4.2 Un fake con i diagrammi degli accordi per chitarra 15
4.3 Partitura per quartetto 16
4.4 Spartito per chitarra 17

5 Considerazioni conclusive **19**

1 Installazione

Come ho detto in premessa, per installare il software dobbiamo ricorrere al source, disponibile all'indirizzo *https://github.com/muparkkra/mup*, seguendo le istruzioni ivi presenti per compilazione su Linux, Windows e Mac.

Questo lavoro, non alla portata di un dilettante, può essere evitato se siamo su Fedora Linux, dove lo possiamo installare dal repository con `dnf` (su Fedora Jam Mup è preinstallato).

All'indirizzo *https://rpmfind.net/linux/rpm2html/search.php?query=application(mupmate.desktop)* potremmo procurarci il file `mup-7.0-1.el8.x86_64.rpm` e installarlo su Debian e derivate, anche di versioni non recenti, previa conversione in `.deb` utilizzando il software Alien [2] .

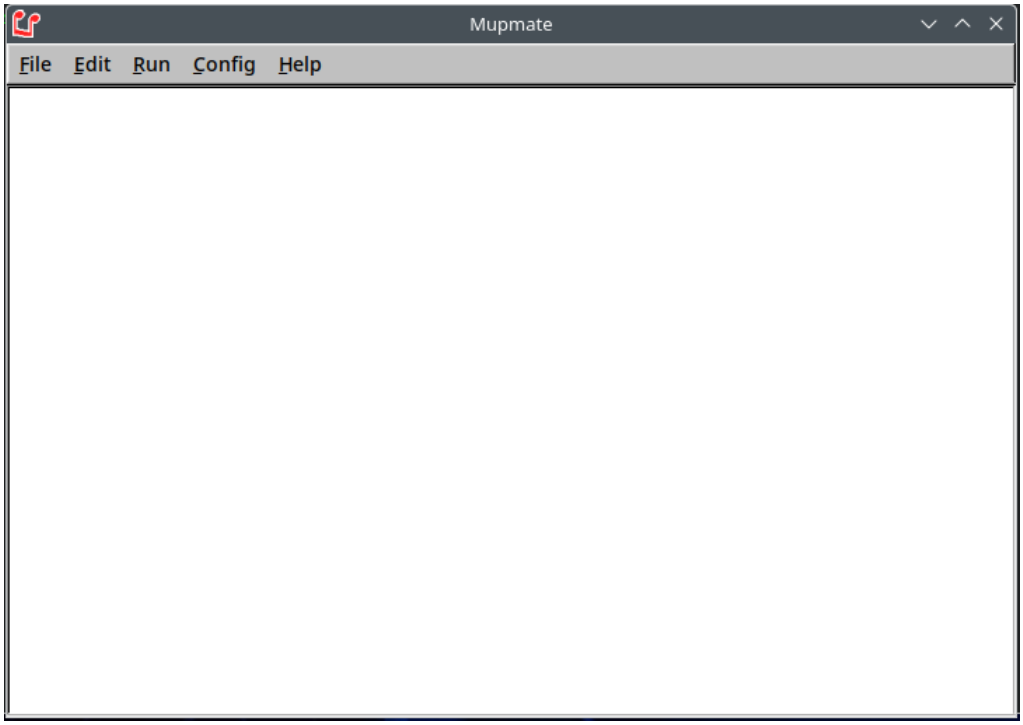
2 Come funziona

Con un editor di testo scriviamo una serie di istruzioni, utilizzando il linguaggio che vedremo, finalizzate a produrre una scrittura musicale e salviamo il file con estensione `.mup`.

Posizionati nella directory che contiene il file, abbiamo a disposizione, tra i tanti altri, i seguenti comandi a terminale che bastano e avanzano per un dilettante:

`mup -f` <nome_file`.ps`> <nome_file`.mup`>
per generare un file postscript dal file mup,
`mup -f` <nome_file`.pdf`> <nome_file`.mup`>
per generare un file PDF dal file mup,
`mup -m` <nome_file`.mid`> <nome_file`.mup`>
per generare un file MIDI dal file mup.

Per facilitarci il lavoro il software contiene un editor, chiamato Mupmate, che si lancia con il comando `mupmate` e si presenta così

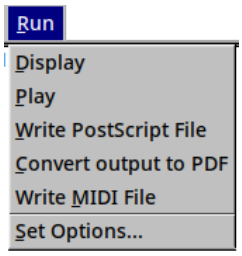


Si tratta di un normale editor di testo il cui menu facilita le cose.

Innanzi tutto da menu HELP ▷ MUP USER'S GUIDE accediamo a un manuale fatto molto bene, purtroppo in lingua inglese, dal quale possiamo trarre approfondimenti a completamento di quanto vedremo in questo manuale, per fare cose più difficili ed elaborate di quelle che vedremo qui.

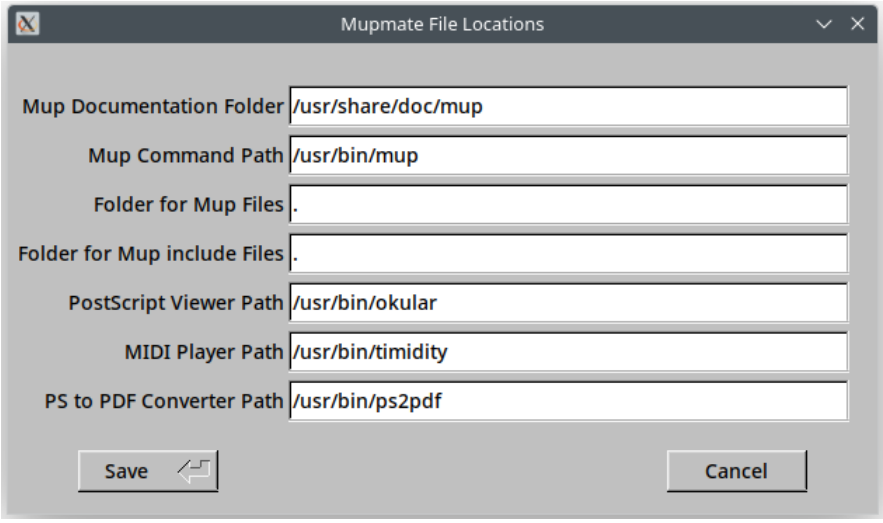
[2] In proposito si veda il post «Convertire packages per Linux» dell'aprile 2025 sul mio blog *www.vittal.it*

Dopo che abbiamo scritto la nostra musica e abbiamo memorizzato il file `.mup` che la contiene, la voce di menu RUN si apre così



offrendoci la possibilità di vedere la musica che abbiamo scritto (DISPLAY), di udirla, per default eseguita da un pianoforte (PLAY), di produrre un file PostScript e salvarlo (WRITE POSTSCRIPT FILE), di produrre un file PDF e salvarlo (CONVERT OUTPUT TO PDF) e di produrre un file MIDI e salvarlo (WRITE MIDI FILE).

Perché tutto funzioni verifichiamo la configurazione aprendo la voce di menu CONFIG ▷ FILE LOCATIONS.... La finestra che si apre dovrebbe essere simile a questa



con indicati i path ai vari strumenti che rendono possibile vedere il manuale (MUP DOCUMENTATION FOLDER che si indica per default), per accedere al software `mup` (MUP COMMAND PATH che si indica per default), per vedere il file PostScript (POSTSCRIPT VIEWER PATH, nel mio caso `okular`), per sentire suonare la musica che abbiamo scritto (MIDI PLAYER PATH, nel mio caso `timidity`) e per convertire il file `.ps` in file `.pdf` (PS TO PDF CONVERTER PATH, nel mio caso `ps2pdf`).

Ovviamente questi ultimi tre strumenti (`okular`, `timidity` e `ps2pdf`), o similari, devono essere installati sul computer.

Le voci di menu FILE e EDIT contengono le solite funzioni che troviamo in qualsiasi normale editor di testo.

3 Linguaggio MUP in sintesi

Il codice MUP si scrive rispettando precise regole sintattiche.

Ha importanza rispettare il maiuscolo e il minuscolo: il linguaggio MUP fa differenza tra c e C.

3.1 Scrittura delle note

Le note vengono scritte con la notazione anglosassone (c d e f g a b per Do Re Mi Fa Sol La Si) in lettere minuscole.

Con la semplice lettera si indicano le note appartenenti all'ottava del Do centrale. Per note più alte si fa seguire alla lettera indicante la nota un simbolo + per ogni ottava a salire. Per note più basse si fa seguire alla lettera indicante la nota un simbolo - per ogni ottava a scendere.

Le alterazioni delle note si ottengono facendo seguire alla lettera indicante la nota il simbolo `#` per il diesis e il simbolo `&` per il bemolle. Con il simbolo `x` si indica il doppio diesis, con il simbolo `&&` si indica il doppio bemolle e con il simbolo `n` si indica il ritorno al naturale.

Per quanto riguarda la durata, essa viene indicata con un numero che si scrive prima della lettera indicante la nota, secondo la seguente tabella

1	semibreve	intero
2	minima	1/2
4	semiminima	1/4
8	croma	1/8
16	semicroma	1/16
32	biscroma	1/32
64	semibiscroma	1/64

Se la durata non viene indicata si assume la durata di default, corrispondente al numero indicato al denominatore dell'indicazione di tempo: in tempo quattro quarti o tre quarti il numero 4, corrispondente alla semiminima.

La nota puntata si ottiene facendo seguire un punto (`.`) al numero indicante la durata.

La nota staccata si scrive anteponendo alla nota l'istruzione `[with .]`.

La nota accentata si scrive anteponendo alla nota l'istruzione `[with ^]`.

La legatura di portamento tra due note si ottiene facendo seguire alla nota da cui inizia la legatura il simbolo `<>`. Per legare più note si fa seguire alla nota da cui inizia la legatura la scritta `ph` e si fa seguire alla nota in cui finisce la legatura la scritta `eph`.

La legatura di valore, possibile solo tra note della stessa altezza, si ottiene facendo seguire alla nota da cui inizia la legatura il simbolo ~.

Dopo la lettera indicante la nota, più o meno corredata dagli altri simboli che la caratterizzano, è necessario mettere un punto e virgola (`;`).

I raggruppamenti irregolari di note si ottengono scrivendo il gruppo di note che compongono il raggruppamento tra parentesi graffe (`{` e `}`), indicando dopo la graffa chiusa il numero di note da raggruppare.
Una terzina di crome si scrive `{8c;8d;8e;}3;`.

Gli abbellimenti si ottengono facendo precedere alla notina di abbellimento l'istruzione `[grace]`. La durata della notina non influisce sulla capienza della misura.

Se invece di indicare una sola nota con la relativa lettera indichiamo più note con le relative lettere scritte di seguito, scriviamo un accordo.
Con l'istruzione `2c;` scriviamo un do della durata di una minima.
Con l'istruzione `4ce&g;` scriviamo una triade di Do minore della durata di una semiminima.

Le pause si inseriscono con la lettera `r` minuscola e la relativa durata si indica esattamente come si fa per le note.

3.2 Scrittura del rigo musicale

Il rigo musicale si scrive dedicando una riga di codice a ciascuna misura.

La riga di codice deve iniziare con un numero indicante il rigo cui ci si riferisce seguito dal simbolo di due punti (`:`) e dalle note da inserire nella misura.

Scritta la riga dedicata alla misura e dato INVIO, sulla riga successiva si indica il tipo di barra che debba chiudere la misura stessa, normalmente identificata dalla parola di codice `bar`.

E così si prosegue per le misure successive.

La barra conclusiva del brano viene indicata con la parola in codice `endbar`.

Se per un qualsiasi motivo non si volessero far vedere le barre in stampa dovremmo indicare la barra con la parola di codice `invisbar`.

Con le parole in codice `repeatstart` e `repeatend` possiamo indicare l'inizio e la fine di eventuali ripetizioni di parti del rigo.

Mup, per default, è predisposto per lavorare sul rigo numero 1, armato in chiave di violino, in tempo 4/4 e in tonalità Do maggiore, cioè senza alterazioni in chiave.

Sicché, per esempio, se vogliamo creare un rigo contenente le prime due misure della canzoncina Fra' Martino campanaro avvalendoci del rigo di default (identificato dal numero 1) basta che scriviamo questo codice

```
1:  4c;4d;4e;4c;
bar
1:  4c;4d;4e;4c;
bar
```

Per brevità, in ciascuna misura, potremmo evitare di indicare la durata delle tre note successive alla prima in quanto la durata di questa resta valida in tutta la riga fino a nuovo avviso.
Addirittura, coincidendo la durata delle note con quella di default (4 in tempo 4/4), potremmo evitare di indicare la durata anche per la prima nota.
Ma con queste semplificazioni è sempre meglio andarci piano.

3.3 Arricchimento del rigo

Il rigo musicale si presenta spesso arricchito di testo e simboli che aggiungono indicazioni alla semplice scrittura delle note: dal testo dell'eventuale canto rappresentato dalle note stesse alle indicazioni dinamiche, alle indicazioni del contesto armonico del brano musicale.

3.3.1 Testo del canto

Quando scriviamo un brano da eseguirsi con il canto generalmente sotto il rigo che indica le note scriviamo le parole da cantare, sillabate in corrispondenza delle note stesse.

Ciò si ottiene scrivendo, per ciascuna misura sotto la riga di istruzioni riguardante le note, il seguente comando

`lyrics 1:  "<testo_sillabato>";`

dove, riferendoci al rigo di default numerato 1, <testo_sillabato> è il testo da cantare diviso nelle sillabe, separate da un semplice spazio, corrispondenti alle note.

3.3.2 Indicazioni dinamiche e varie

Vengono scritte sopra il rigo (per esempio per indicare il tempo di metronomo) o sotto il rigo (per esempio per indicare i crescendo i diminuendo, i piano, i forte, ecc.).

La sintassi del comando per ottenere queste scritte è la seguente

<forma_font> <posizione> <rigo>: <battuta> "<testo>";

dove:

<forma_font> può essere
- `rom` per un carattere normale,
- `ital` per un carattere inclinato,
- `bold` per un carattere grassetto,
- `boldital` per un carattere inclinato grassetto,

<posizione> può essere
- `above` per ottenere la scritta sopra il rigo,
- `below` per ottenere la scritta sotto il rigo,

<rigo>
- è il numero del rigo (1 per il rigo di default),

<battuta>
- è il numero che indica la battuta della misura in corrispondenza alla quale deve apparire la scritta (il numero `0` corrisponde all'inizio della misura),

<testo> è il testo della scritta.

La parte del comando <battuta> "<testo>" è ripetibile sulla stessa riga, ovviamente nei limiti del numero di battute contenute nella misura, separata dalla precedente con un punto e virgola.

Esempi:
`rom above 1: 0 "Presto";` fa comparire la scritta Presto sopra il rigo all'inizio della misura del rigo 1 in cui è inserito il comando, utilizzando un carattere normale,
`ital below 1: 3 "Crescendo";` fa comparire la scritta Crescendo sotto il rigo in corrispondenza della terza battuta della misura del rigo 1 in cui è inserito il comando, utilizzando un carattere inclinato.

3.3.3 Indicazioni armoniche

Come avviene nei così detti fake book, molto utilizzati nella musica folk, come traccia per l'esecuzione solistica di un brano, e nel jazz, come traccia per le improvvisazioni ispirate alla melodia e all'armonia del brano, il rigo è arricchito dalle sigle degli accordi che costituiscono la struttura armonica del brano, generalmente scritte sopra il rigo.

L'inserimento di questo elemento avviene con il comando
<forma_font> `chord` <rigo>`:` <battuta> `"`<sigla>`";`
dove per <forma_font>, <rigo> e <battuta> vale quanto detto nel precedente paragrafo e <sigla> è il nome della tonica in maiuscolo seguito da eventuale accidente (`#` per diesis, `&` per bemolle) e dall'eventuale modificatore in minuscolo: il solo nome della tonica senza modificatori rappresenta l'accordo maggiore.
I modificatori sono quelli comunemente utilizzati dai musicisti: `m` per l'accordo minore, `7` per l'accordo di settima di dominante, `M7` per l'accordo di settima maggiore, `aug` per l'accordo aumentato, `dim` per l'accordo diminuito, ecc.
Il simbolo Δ che ricorre per esprimere gli accordi di settima maggiore si ottiene con il tasto ^ .
Il simbolo $^\circ$ che ricorre negli accordi di settima diminuita si ottiene scrivendo `o:`.

La parte del comando <battuta> "<sigla>" è ripetibile sulla stessa riga, ovviamente nei limiti del numero di battute contenute nella misura, separata dalla precedente con un punto e virgola.

3.3.4 Basso continuo

Per le indicazioni relative all'esecuzione del basso continuo, detto anche basso figurato o numerato, generalmente poste sotto il rigo, abbiamo a disposizione il comando
<forma_font> `figbass` <rigo>`:` <battuta> `"`<contenuto>`";`
dove per <forma_font>, <rigo> e <battuta> vale quanto detto nei precedenti paragrafi e <contenuto> riguarda il numero o i numeri o i simboli da esporre, tenendo presente che, per i numeri sovrapposti, il primo che si indica è quello che sta sopra.

Per default le scritte per il basso continuo vengono poste sotto il rigo. Se si volessero far apparire sopra il rigo occorrerebbe far seguire alla parola chiave `figbass` la parola chiave `above`.

La parte del comando <battuta> "<contenuto>" è ripetibile sulla stessa riga, nei limiti del numero di battute contenute nella misura, separata dalla precedente con un punto e virgola.

Esempio:
`rom figbass 1: 1 "4"; 2 "#"; 3 "6 4"; 4 "7";`
produce, sotto il rigo, rispettivamente in corrispondenza delle battute 1, 2, 3 e 4 i simboli

$4 \qquad \sharp \qquad \begin{matrix}6\\4\end{matrix} \qquad 7\cdot$

3.3.5 Indicazioni di analisi armonica

Nei trattati di armonia si usa qualificare gli accordi riferendosi ai gradi della scala su cui sono costruiti e si indicano con la particolare simbologia ben nota a chi studia armonia.

Per creare queste indicazioni, generalmente poste sotto il rigo, abbiamo a disposizione il comando

<forma_font> `analysis` <rigo>: <battuta> "<contenuto>";
dove per <forma_font>, <rigo> e <battuta> vale quanto detto nei precedenti paragrafi e <contenuto> riguarda i simboli da esporre, da indicare secondo la seguente sintassi:
. se il simbolo riguarda semplicemente il grado della scala se ne indica il numero romano tra i doppi apici,
. se il simbolo è costituito dal numero romano arricchito da qualificatori alla sua destra in pedice e apice, si indica il grado della scala con il relativo numero romano seguito da due punti (:) e, immediatamente dopo i due punti si scrive quanto da far apparire in apice e, separato da uno spazio, si scrive quanto da far apparire in pedice.

Per default queste scritte vengono poste sotto il rigo. Se si volessero far apparire sopra il rigo occorrerebbe far seguire alla parola chiave `analysis` la parola chiave `above`.

La parte del comando <battuta> "<contenuto>" è ripetibile sulla stessa riga, ovviamente nei limiti del numero di battute contenute nella misura, separata dalla precedente con un punto e virgola.

Esempio:
`rom analysis 1:  1 "I"; 2 "II: 4"; 3 "V:&6 #4"; 4 "III:6 4";`
produce, sotto il rigo, rispettivamente in corrispondenza delle battute 1, 2, 3 e 4 i simboli
I II_4 $\mathrm{V}^{b6}_{\#4}$ III^6_4.

3.4 Spartiti e partiture

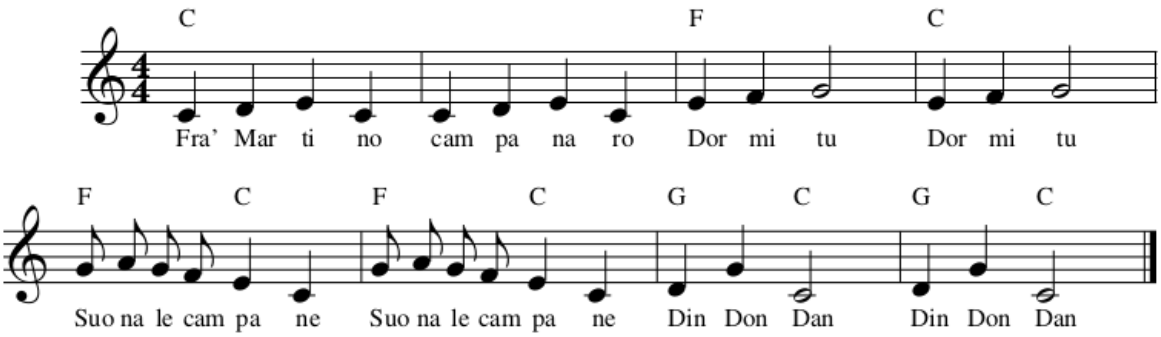
Con ciò che abbiamo visto nei precedenti paragrafi saremmo in grado di produrre un rigo musicale, più o meno arricchito, avvalendoci del rigo di default numerato con il numero 1, in chiave di Sol, in tempo 4/4 e in tonalità di Do maggiore.

Per esempio, con questo script

```
1:  4c;d;e;c;
lyrics 1:  "Fra' Mar ti no";
rom chord 1:  1 "C";
bar
1:  4c;d;e;c;
lyrics 1:  "cam pa na ro";
bar
1:  4e;f;2g;
lyrics 1:  "Dor mi tu";
rom chord 1:  1 "F";
bar
1:  4e;f;2g;
lyrics 1:  "Dor mi tu";
rom chord 1:  1 "C";
bar
1:  8g;a;g;f;4e;c;
lyrics 1:  "Suo na le cam pa ne";
rom chord 1:  1 "F"; 3 "C";
bar
1:  8g;a;g;f;4e;c;
lyrics 1:  "Suo na le cam pa ne";
rom chord 1:  1 "F"; 3 "C";
bar
1:  4d;g;2c;
lyrics 1:  "Din Don Dan";
rom chord 1:  1 "G"; 3 "C";
bar
1:  4d;g;2c;
```

```
lyrics 1:  "Din Don Dan";
rom chord 1:  1 "G"; 3 "C";
endbar
```

produciamo questo



Un bel rigo arricchito che presenta due difetti: manca il titolo e, almeno per i perfezionisti, le crome contigue sono slegate tra loro, quando la scrittura musicale ci ha abituati a vederle raggruppate da una barretta.

Per rimediare abbiamo bisogno di scrivere alcune istruzioni nel nostro script, istruzioni che hanno un valore preliminare alla scrittura della musica e che fanno parte di un contesto diverso da quello che ho descritto nei paragrafi precedenti.

Per produrre uno spartito o una partitura completi e svincolati da quelle che sono le impostazioni di default (un solo rigo più o meno arricchito ma senza titoli, in tonalità di Do maggiore, in tempo 4/4 e adatto per l'inserimento di una sola voce) abbiamo bisogno di quelli che nel linguaggio MUP si chiamano contesti. I contesti principali sono:
`header`, dove indichiamo titoli per la prima pagina,
`header2`, dove indichiamo titoli per le pagine successive,
`footer`, dove indichiamo scritte da porre a piede della prima pagina,
`footer2`, dove indichiamo scritte da porre a piede delle pagine successive,
`score`, dove indichiamo parametri da applicare all'intera partitura,
`staff n`, dove indichiamo parametri da applicare al rigo n (con n da 1 a 40),
`music`, dove indichiamo ciò di cui abbiamo parlato nei paragrafi precedenti.

Nelle righe successive a quella contenente la parola chiave che identifica il contesto inseriamo le istruzioni relative al contesto stesso. Qui di seguito indico la sintassi delle istruzioni più ricorrenti per ciascun contesto.

header e footer

`title` <forma_font> `(n)` "<testo>" "<testo>"

inserisce un titolo.
Per <forma_font> rimando a quanto detto nel paragrafo 3.3.2.
In mancanza di questa indicazione per default viene usato un carattere normale.
(n) indica la dimensione del font.
Anche questa indicazione è facoltativa. Per default viene usato corpo 11.
<testo> è il testo da scrivere.
Se l'indicazione è una sola il testo viene scritto centrato.
Se inseriamo due indicazioni, la prima viene scritta a sinistra e la seconda a destra.

score

`staffs =` <n>

stabilisce il numero <n> di righi che compongono una partitura.
Questo numero può variare da 1 a 40 (numero massimo di righi per MUP).

Per spartiti che non siano per pianoforte o organo l'indicazione non serve.

`key =` <indicazione>

stabilisce la tonalità della partitura o dello spartito.

<indicazione> può essere un nome o il numero delle alterazioni in chiave.

Sol maggiore si può indicare con G major o con 1#.

Re minore si può indicare con D minor o con 1&.

`time =` <indicazione>

stabilisce il tempo, cioè le battute che compongono la misura.

<indicazione> è la frazione che normalmente si usa (4/4, 6/8, 3/4, ecc.).

Chi ama l'antico può indicare `common` per 4/4 (C comune) o `cut` per 2/2 (C tagliato).

`beamstyle =` <indicazione>

la beam è la linea con cui si usano unire tra loro successive note di durata di 1/8 o inferiore.

<indicazione> stabilisce il tempo occupato dalle note unite:

con `4,4,4,4` si uniscono note fino a formare la durata di 1/4,

con `2,2` si uniscono note fino a formare la durata di 1/2.

`bracket =` <n> `-` <n> `("`<nome>`")`

pone una parentesi quadra sulla sinistra dei due righi contigui <n> e <n>.

Opzionalmente se ne può indicare il nome da scrivere sulla sinistra della parentesi.

`brace =` <n> `-` <n> `("`<nome>`")`

pone una parentesi graffa sulla sinistra dei due righi contigui <n> e <n>.

Opzionalmente se ne può indicare il nome da scrivere sulla sinistra della parentesi.

staff n

`clef =` <nome>

indica la chiave da usare per il rigo n.

<nome> può essere

`treble` per la chiave di violino

`treble8` per la chiave di violino che evidenzi la trasposizione di una ottava (chitarra)

`bass` per la chiave di basso

`soprano`

`mezzosoprano`

`alto`

`tenor`

`baritone`

`label = ("`<nome>`")`

indica un nome da scrivere sulla sinistra del rigo n, come si usa fare nelle partiture.

<nome> è la stringa con il nome di uno strumento.

`vscheme =` <n><opzione>

indica le voci che si possono scrivere sullo stesso rigo (rigo polifonico).

<n> può essere 1 2 o 3 (3 è il numero massimo di voci che MUP gestisce sul rigo)

<opzione> può essere

`o` se le stanghette delle note devono essere contrapposte,

`f` se le stanghette delle note possono avere orientamento libero.

In questo contesto è possibile inserire il comando `beamstyle` che ho descritto nel contesto `score` precedentemente trattato.

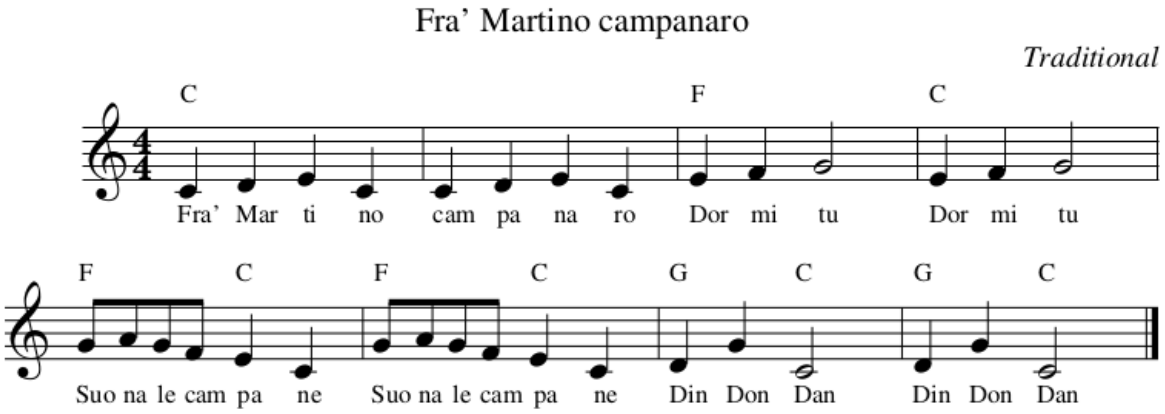
music

In questo contesto inseriamo tutto ciò che abbiamo visto nei paragrafi 3.1, 3.2 e 3.3.

Tornando al nostro esempio di Fra' Martino campanaro, per avere un fake più completo e stilisticamente secondo usanza, tenendo conto di quanto abbiamo sopra indicato, dobbiamo scrivere, prima dello script indicato alle pagine 8 e 9 le seguenti righe di codice

```
header
title (16) "Fra' Martino campanaro"
title ital (14) "" "Traditional"
score
beamstyle = 2,2
music
```

ed otteniamo in stampa



3.5 Riservato ai chitarristi

Il linguaggio MUP fornisce ai chitarristi la possibilità di scrivere musica attraverso l'intavolatura e di indicare gli accordi sopra il rigo con il loro diagramma, cioè con il loro schema esecutivo.

3.5.1 Intavolatura

L'intavolatura serve per scrivere le note indicandone la posizione su un manico di chitarra stilizzato in sei righe parallele orizzontali.

Il risultato in stampa sarà un rigo indicante le note nella consueta notazione musicale sovrapposto all'intavolatura così creata, che aiuta il chitarrista principiante a trovare la posizione delle dita della mano sinistra per eseguire le note stesse.

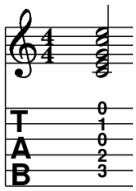
Per predisporci ad ottenere tutto ciò occorre:
. collocare nel contesto `score` l'istruzione `staff=2` in modo da prevedere due righi,
. collocare nel contesto `staff 2` l'istruzione `stafflines=tab`
Quest'ultima istruzione è una istruzione abbreviata che genera per default l'intavolatura per la chitarra ma MUP può generare intavolature anche per altri strumenti a corda, fino a 9 corde. L'istruzione completa per generare l'intavolatura per qualsiasi strumento è
`stafflines=tab(<lista_delle_corde>)`
dove <lista_delle_corde> è una lista dei nomi delle corde seguiti dall'ottava di appartenenza della corrispondente nota generata.
Per costruire intavolature per strumenti diversi dalla chitarra (banjo, mandolino, chitarra basso, ukulele, ecc.) ci si può regolare sapendo che l'istruzione riferita alla chitarra, che opzionalmente possiamo usare al posto di quella abbreviata che ho indicato prima, sarebbe
`stafflines=tab(e5 b4 g4 d4 a3 e'3)`
dove abbiamo le corde nominate con i nomi della nota generata a corda libera, a partire dal cantino `e` per arrivare alla corda bassa `e'`, così nominata per distinguerla dall'altra, seguiti dall'ottava di appartenenza della nota generata a corda libera (5 per il cantino, in quanto genera a corda libera, nella notazione trasposta della chitarra, un Mi appartenente alla quinta ottava, quella del Do centrale, e un 3 per la corda bassa in quanto genera a corda libera un Mi appartenente alla terza ottava).

La prima indicazione serve per identificare la corda nella scrittura delle note attraverso l'intavolatura e la seconda indicazione serve a MUP per scrivere nella giusta posizione la nota nel rigo musicale generato sopra l'intavolatura.

Un esempio per capire tutto il meccanismo.
Con questo script

```
score
staffs=2
staff 2
stafflines=tab
music
2:2a3d2g0b1e0;
bar
```

generiamo questo accordo di Do maggiore con la sua intavolatura



Come si vede, indicando, nel contesto `music` per il rigo 2, dopo la durata di minima con il numero 2, i nomi delle corde seguiti dal numero della tacca su cui premere la corda (0 per la corda libera), MUP genera sul rigo 2 l'intavolatura e, sopra, sul rigo 1 di default, la notazione musicale di ciò che si è scritto nell'intavolatura.

3.5.2 Diagrammi degli accordi

Un altro modo di indicare dove mettere le dita della mano sinistra sul manico della chitarra per generare un accordo è quello dei diagrammi che, nei righi con indicati i riferimenti armonici dei brani, si usano mettere sotto il nome degli accordi, a vantaggio dei chitarristi dilettanti.

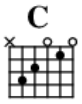
Mup ci offre la possibilità di costruire questi diagrammi in un contesto chiamato `grids` con l'istruzione
`"<sigla>" "<tacche>"`
dove
<sigla> è la stessa indicazione che abbiamo visto nel paragrafo 3.3.3 per l'indicazione degli accordi in arricchimento del rigo,
<tacche> è la lista delle tacche del manico della chitarra da premere, partendo dalla corda bassa.

Per predisporre il diagramma per lo stesso accordo di Do maggiore visto nell'esempio alla fine del paragrafo precedente l'istruzione è
`"C" "x 3 2 o 1 o"`
dove, per ciascuna corda, a partire dalla sesta (il Mi basso), con `x` si indica la corda da non utilizzare, con `o` si indica la corda da utilizzare libera e con il numero si indica la tacca su cui premere il dito della mano sinistra.

Nel contesto `grids` si predispongono tutti gli accordi da utilizzare nel brano e il loro utilizzo si predispone inserendo nel contesto score l'istruzione
`gridswhereused=y`

Fatto questo, quando, nel contesto `music`, usiamo l'istruzione per scrivere l'accordo, insieme al nome dell'accordo verrà riprodotto il suo diagramma.

Nel caso dell'esempio appena fatto



Lo stesso procedimento può essere seguito per qualsiasi strumento con un numero di corde diverso da quello della chitarra indicando nella lista delle tacche quattro elementi anziché sei.

Per esempio, il più semplice accordo eseguibile sull'ukulele, quello del primo rivolto di Do maggiore, si predispone con

```
"C/G" "o o o 3"
```

3.6 Gestione del MIDI

Abbiamo visto nel Capitolo 2 che MUP può farci sentire ciò che abbiamo scritto per produrre spartiti e partiture e può addirittura generare un file MIDI.

Per default, se non inseriamo nel nostro script alcuna istruzione MIDI, la riproduzione avviene con il suono del pianoforte in tempo di metronomo 120 (tra l'Allegro moderato e l'Allegro) e velocity 64 (volume medio nel range 0 - 127).

Ovviamente il linguaggio MUP prevede istruzioni per modificare tutto questo.

Queste istruzioni si piazzano all'inizio del contesto `music`, appena prima di cominciare l'inserimento delle note.

Per regolare il tempo di metronomo l'istruzione è

```
midi all:  0 "tempo=<n>";
```

dove <n> è il tempo di metronomo voluto.
Il termine `all` sta ad indicare che l'istruzione riguarda tutti i righi (staff).

Le altre istruzioni devono essere riferite a ciascun rigo e si introducono con `midi 1`, `midi 2`, ecc. dove i numeri sono quelli degli staff.

Le istruzioni vere e proprie sono poi

```
0 "channel=<n>";
```

per indicare il canale MIDI con n da 1 a 16 (10 riservato alle percussioni),

```
0 "program=<n>";
```

per indicare lo strumento (n è il numero dello strumento secondo lo standard general MIDI),

```
0 "onvelocity=<n>";
```

per indicare la velocity, cioè il volume del suono (n varia da 0 a 127).

Se si utilizza un solo strumento è inutile indicare il canale.

Se si utilizzano più strumenti su diversi righi ogni strumento deve avere un suo canale.

Lo 0 che precede i comandi si può sostituire con il numero della battuta da cui debba avere effetto il comando: lo zero indica l'inizio della misura.

4 Alcuni esempi

Con quanto ho illustrato in questo manualetto possiamo fare tante cose, anche abbastanza complicate, come dimostro in questi esempi.

4.1 Spartito per pianoforte con analisi armonica

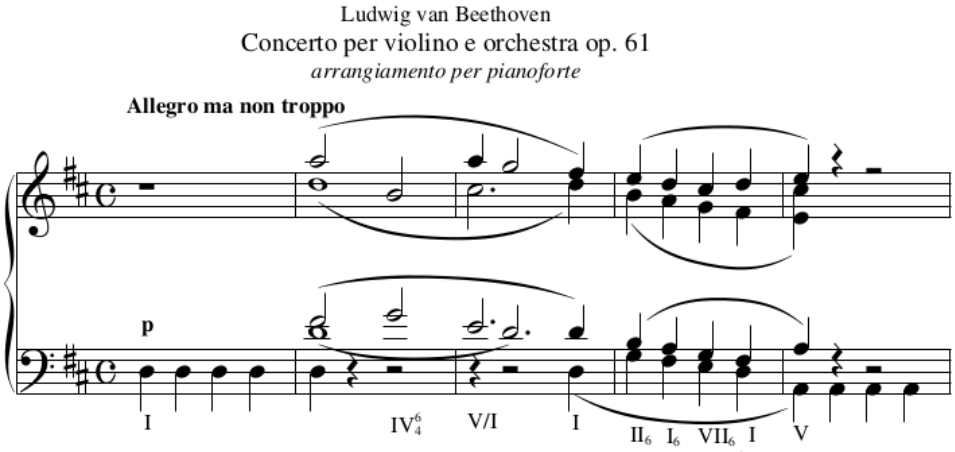
Con questo script vediamo come si realizza il doppio rigo per il pianoforte e come si possono evidenziare elementi di analisi armonica:

```
header
    title (12) "Ludwig van Beethoven"
    title (14) "Concerto per violino e orchestra op.  61"
    title ital (12) "arrangiamento per pianoforte"
score
    staffs = 2
    brace = 1-2
```

```
    key = 2#
    time = common
staff 1
    vscheme = 2o
staff 2
    clef = bass
    vscheme = 3f
music
    bold above 1:  0 "Allegro ma non troppo";
    1 2:  1r;
    bold above 2:  1 "p";
    2 2:  4d;d;d;d;
    rom analysis 2:  1 "I";
    bar
    1 1:  2a+ph;b;
    1 2:  1d+ph;
    2 1:  2f+ph;g+;
    2 3:  1d+ph;
    2 2:  4d;4r;2r;
    rom analysis 2:  3 "IV:6 4";
    bar
    1 1:  4a+;2g+;4f+eph;
    1 2:  2.c+;4d+eph;
    2 1:  2.e+;4d+eph;
    2 3:  2.d+eph;
    2 2:  4r;2r;4dph;
    rom analysis 2:  1 "V/I"; 4 "I";
    bar
    1 1:  4e+ph;d+;c+;d+;
    1 2:  4bph;a;g;f;
    2 1:  4bph;a;g;f;
    2 2:  4g;f;e;d;
    rom analysis 2:  1 "II: 6"; 2 "I: 6"; 3 "VII: 6"; 4 "I";
    bar
    1 1:  4e+eph;4r;2r;
    1 2:  4ec+eph;
    2 1:  4aeph;4r;2r;
    2 2:  4a-eph;a-;a-;a-;
    rom analysis 2:  1 "V";
    bar
```

Le indentazioni non sono necessarie. Le ho utilizzate per mostrare meglio il contenuto dei vari contesti.

La stampa che otteniamo con questo script è la seguente



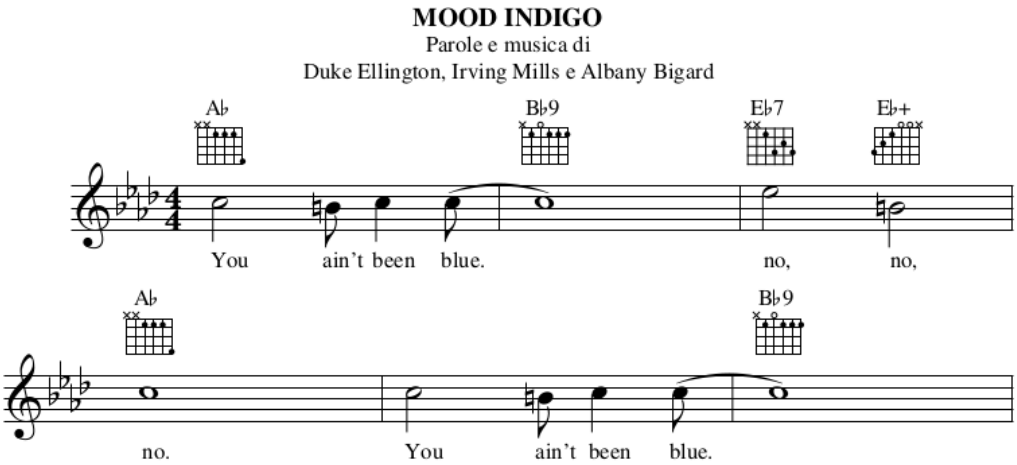
4.2 Un fake con i diagrammi degli accordi per chitarra

Con questo script realizziamo uno spartito del tipo molto utilizzato per la musica leggera e per la musica jazz.

Per dare un aiuto ad un chitarrista alle prime armi le indicazioni armoniche sono corredate da diagrammi che mostrano come ottenere gli accordi sulla chitarra.

```
header
    title bold (14) "MOOD INDIGO"
    title rom (12) "Parole e musica di"
    title rom (12) "Duke Ellington, Irving Mills e Albany Bigard";
score
    key = 4&
    gridswhereused=y
grids
    "A&" "x x 1 1 1 4"
    "B&9" "x 1 o 1 1 1"
    "E&7" "x x 1 3 2 3"
    "E&+" "3 2 1 o o x"
music
    1:  2c+;8bn;4c+;8c+~;
    rom chord 1:  1 "A&";
    lyrics 1:  "You ain't been blue.";
    bar
    1:  1c+;
    rom chord 1:  1 "B&9";
    bar
    1:  2e+;2bn;
    rom chord 1:  1 "E&7"; 3 "E&+";
    lyrics 1:  "no, no,";
    bar
    1:  1c+;
    rom chord 1:  1 "A&";
    lyrics 1:  "no.";
    bar
    1:  2c+;8bn;4c+;8c+~;
    lyrics 1:  "You ain't been blue.";
    bar
    1:  1c+;
    rom chord 1:  1 "B&9";
    bar
```

In stampa otteniamo



4.3 Partitura per quartetto

Ora realizziamo una partitura per un quartetto. Con la stessa tecnica possiamo realizzare una partitura per una intera orchestra sinfonica.

Lo script è questo

```
header
    title (14) "Ludwig van Beethoven"
    title (16) "Quartetto in Mi bemolle maggiore WoO 36, n.  1"
score
    staffs = 5
    key = e& major
    time = 2/4
    brace = 4-5 ("Pianoforte")
    beamstyle = 4,4
staff 1
    label = "Violino"
staff 2
    label = "Viola"
    clef = alto
staff 3
    label = "Violoncello"
    clef = bass
staff 5
    clef = bass
music
    midi all:  0 "tempo=50";
    midi 1:  0 "channel=1"; 0 "program=40";
    midi 2:  0 "channel=2"; 0 "program=41";
    midi 3:  0 "channel=3"; 0 "program=42";
    midi 4-5:  0 "channel=4"; 0 "program=0"; 0 "onvelocity=127";
    1:  4b-g;8b-g;8b-g;
    bold above 1:  0 "Adagio assai";
    boldital below 1:  1 "p";
    2:  4g-e;8g-e;8g-e;
    boldital below 2:  1 "p";
    3:  4e;8e;8e;
    boldital below 3:  1 "p";
    4:  4eg;8eg;16.g;32a;
    boldital below 4:  1 "p";
    5:  4e;8e;8e;
    bar
    1:  8ce;4cf;8cg;
    boldital below 1:  1.5 "f"; 2.5 "p";
    2:  8a-;4a-;8a-;
    boldital below 2:  1.5 "f"; 2.5 "p";
    3:  4a-;4a-;
    boldital below 3:  2 "fp";
    4:  16g;16f;4c+;16a;16f;
    boldital below 4:  1.5 "f"; 2.5 "p";
    5:  4a-a;4a-a;
    boldital below 5:  2 "fp";
    bar
    1:  8g-e;8b;4d;
    2:  8b-;8g-;4a-;
    3:  4b-;4b-n;
    4:  4g-e;[grace]16e;[grace]16f;[grace]16g;8.a-f;16e;
    5:  4b-;4b-n;
    bar
```

Per rispettare l'indicazione di Adagio assai ho portato a 50 il tempo di metronomo per il MIDI.

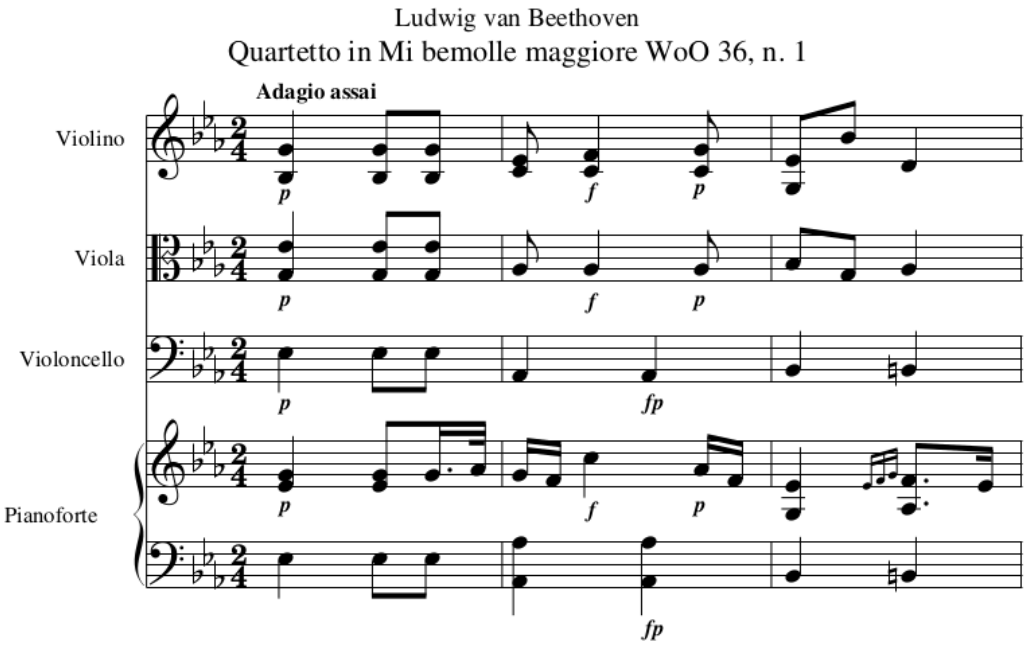
Ho poi assegnato gli strumenti appropriati ai vari righi.

Il tutto affinché l'audizione MIDI rispecchi il più fedelmente possibile la partitura.

Notare come l'indicazione delle note debba tener conto, per identificare l'ottava di appartenenza e per la giusta collocazione dei + e dei -, del tipo di chiave utilizzato per il rigo e della posizione che in quel rigo ha il Do centrale.

Notare anche la possibilità di identificare la posizione nella misura per inserire scritte possa utilizzare anche numeri decimali per posizioni intermedie a quelle scandite dal tempo in chiave.

In stampa otteniamo



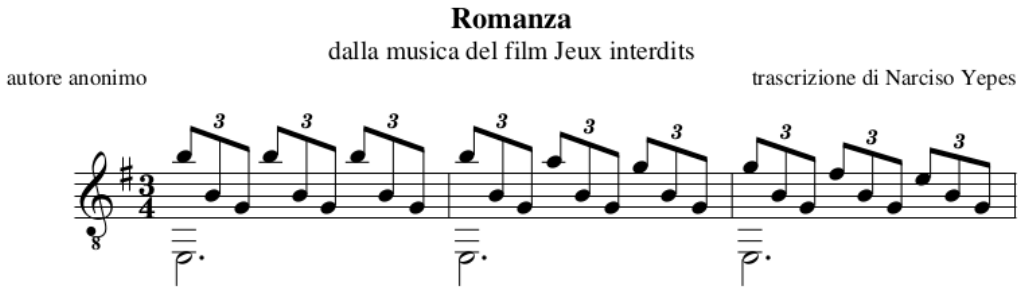
4.4 Spartito per chitarra

Con questo script produciamo un normale spartito per chitarra.

```
header
    title bold (16) "Romanza"
    title rom (14) "dalla musica del film Jeux interdits"
    title rom (12) "autore anonimo" "trascrizione di Narciso Yepes"
score
    key = e minor
    time = 3/4
    beamstyle = 4,4,4
staff 1
    clef = treble8
    vscheme = 2f
music
    midi 1:  0 "program=24";
    1 1:  {8b+;b;g;}3;{b+;b;g;}3;{b+;b;g;}3;
    1 2:  2.e-;
    bar
    1 1:  {8b+;b;g;}3;{a+;b;g;}3;{g+;b;g;}3;
    1 2:  2.e-;
    bar
    1 1:  {8g+;b;g;}3;{f+;b;g;}3;{e+;b;g;}3;
    1 2:  2.e-;
    bar
```

Notare come, per poter inserire sullo stesso rigo note di diversa durata nella stessa battuta, dobbiamo ricorrere alla tecnica di scrittura polifonica, prevedendo più voci per lo stesso rigo.

In stampa otteniamo



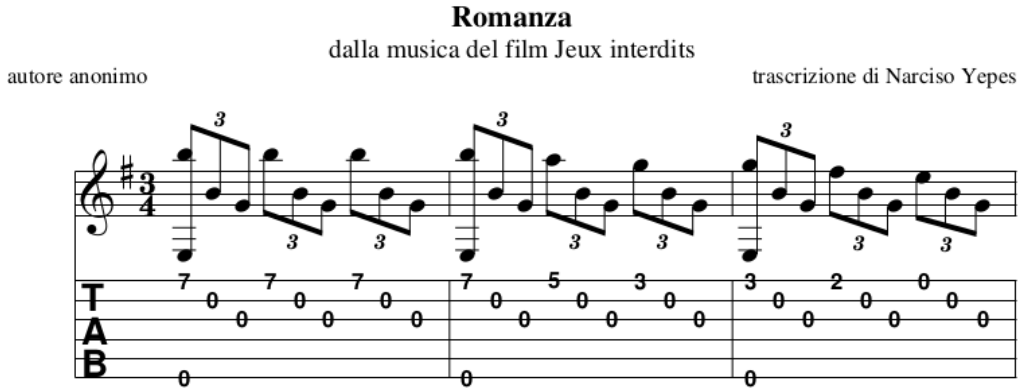
Quest'altro script produce lo stesso spartito inserendo le note attraverso l'intavolatura, in modo che il chitarrista principiante sia guidato nell'esecuzione.

```
header
    title bold (16) "Romanza"
    title rom (14) "dalla musica del film Jeux interdits"
    title rom (12) "autore anonimo" "trascrizione di Narciso Yepes"
score
    key = e minor
    time = 3/4
    beamstyle = 4,4,4
staff 2
    stafflines = tab
music
    midi 1:  0 "program=24";
    2:  {8e'0e7;b0;g0;}3;{8e7;b0;g0;}3;{8e7;b0;g0;}3;
    bar
    2:  {8e'0e7;b0;g0;}3;{8e5;b0;g0;}3;{8e3;b0;g0;}3;
    bar
    2:  {8e'0e3;b0;g0;}3;{8e2;b0;g0;}3;{8e0;b0;g0;}3;
    bar
```

Molto simile allo script precedente, con la differenza che le note sono inserite indicandone la posizione sul manico della chitarra attraverso l'intavolatura e manca lo stile polifonico in quanto esso non è realizzabile utilizzando l'intavolatura (le note nella stessa posizione devono avere la stessa durata, come quando si scrivono gli accordi).

In entrambi i casi ho fatto in modo che la riproduzione MIDI avvenga con il suono della chitarra classica.

In stampa otteniamo



5 Considerazioni conclusive

Penso che da questo manualetto si siano acquisiti elementi per fare parecchie cose con MUP impiegando meno tempo di quello che si sarebbe impiegato affidandosi alla User's Guide che ci propone il menu Help di Mupmate.

Chi voglia saperne di più per fare cose più complicate di quelle affrontate qui si può affidare a quella guida.

Nei file di documentazione che si installano installando MUP (che in un filesystem Linux si trovano in /usr/share/doc/mup) troviamo anche il file `sample.mup` che costituisce un utile prontuario dei vari comandi contemplati dal linguaggio.

Ovviamente occorre stampare il contenuto di testo di questo file e compilare il file stesso con il comando a terminale

```
mup -f sample.pdf sample.mup
```

per vedere l'effetto dei vari comandi descritti nel testo.