



alev

FidoCadJ portabile, un CAE al guinzaglio

8 November 2011

Indice

- [1 Introduzione](#)
- [2 Primo passo - il dispositivo USB](#)
- [3 Secondo passo - Java Runtime Environment \(JRE\)](#)
- [4 Terzo passo - FidoCadJ](#)
- [5 Quarto passo - gli scripts di attivazione](#)
- [6 Prova di funzionamento](#)
- [7 Approfondimenti](#)

[\[modifica\]](#) Introduzione

Nel gergo tecnico in campo informatico, un'applicazione si definisce portabile quando la stessa può essere eseguita senza installarla sul disco fisso di un computer, bensì mediante l'utilizzo di un dispositivo rimovibile, tipicamente una **chiavetta USB**.

Nota: la definizione di *applicazione portabile* non deve essere confusa con quella di [software portabile](#).

I vantaggi di questo tipo di memorizzazione sono vari:

- l'applicazione è sempre disponibile, anche su computer differenti, a parità di sistema operativo
- disponendo di diversi computer, non si è costretti ad installare su tutti i computer l'applicazione e neppure le eventuali dipendenze
- dovendo spostarsi di luogo, è possibile portarsi dietro la sola chiavetta, purchè nella destinazione risulti disponibile un computer con il medesimo sistema operativo.

Partecipando alle varie discussioni su ElectroYou, capita con una certa frequenza che un utente non possa inserire schemi nel forum perché sul computer in uso in quel momento non è possibile installare applicazioni; io stesso, usando al lavoro un computer aziendale

con utenza non privilegiata, incappo spesso nel problema di non poter modificare gli schemi circuitali fatti con FidoCadJ quando sono a casa.

Partendo da problemi di questo tipo, ho pensato di rendere FidoCadJ portabile su una chiavetta USB, compito facilitato dal fatto che questa applicazione è sviluppata in linguaggio Java® su un singolo jar contenente tutte le componenti software dell'applicazione; sulla chiavetta è previsto anche di memorizzare una copia del JRE (Java Runtime Environment) che è il *motore software*, specifico del sistema operativo, che permette di eseguire il jar di FidoCadJ su un generico computer.

La procedura nel seguito descritta è differenziata tra Microsoft Windows® XP e Linux; il medesimo jar di FidoCadJ funziona su entrambi i sistemi operativi con la sola condizione che sulla chiavetta sia memorizzato il JRE specifico del sistema operativo.

[[modifica](#)] Primo passo - il dispositivo USB

Prima di tutto è necessario disporre di una chiavetta USB di capacità sufficiente; per il JRE di Windows e FidoCadJ sono sufficienti 100 MB ma è bene considerare dello spazio aggiuntivo per custodire gli schemi; se si vuole usare FidoCadJ anche su Linux è necessario memorizzare anche JRE per Linux, che richiede ulteriori 120 MB circa.

La chiavetta è bene che sia formattata in FAT32, in modo da poter leggere e scrivere sia con Windows che con Linux; in alternativa, è possibile creare sulla chiavetta due partizioni, una in FAT32 e l'altra di tipo linux da formattare in ext3 ma, in questo modo, la configurazione è più complicata e la lascio ai più smaliziati.

Nel prosieguo, si farà sempre riferimento ad una installazione su unica partizione FAT32.

E' opportuno che sulla chiavetta sia presente almeno una cartella principale adibita a contenere le varie applicazioni, come FidoCadJ, e/o i software *di supporto*, come il JRE.

A questo punto inseriamo la chiavetta nella porta USB ed attendiamo che venga montata dall'automount (se l'utente lo ha disabilitato, dovrebbe sapere anche come montare la chiavetta manualmente) e, ipotizzando che la chiavetta sia disponibile sull'unità **R:** di Windows oppure montata in **/media/disc** su Linux, creiamo al suo interno la struttura destinata a contenere le componenti software:

- su Windows:

```
md R:\Programmi\FidoCadJ_portable
```

```
md R:\Programmi\Win_JRE
```

```
md R:\Programmi\Linux_JRE
```

- su Linux:

```
mkdir -p /media/disc/Programmi/FidoCadJ_portable  
  
mkdir /media/disc/Programmi/Win_JRE  
  
mkdir /media/disc/Programmi/Linux_JRE
```

Naturalmente, nel caso di Windows, è possibile creare la struttura mediante **Explorer**; in generale, è necessaria l'unica accortezza di non inserire spazi nei nomi delle directories per evitare potenziali *incomprensioni* da parte di Linux.

Terminate le suddette operazioni, lasciamo la chiavetta montata e procediamo con i passi successivi.

[[modifica](#)] Secondo passo - Java Runtime Environment (JRE)

Come già anticipato, il JRE provvede alla funzione di motore di un'applicazione scritta in linguaggio java; per ogni comune sistema operativo (Microsoft Windows, Linux, Mac OS e via dicendo) esiste uno specifico JRE sviluppato dalla Oracle Corporation, proprietaria della tecnologia e del marchio.

Per disporre del JRE sulla chiavetta, è necessario procedere come segue:

- [scaricare il programma di installazione](#) e lanciarne l'esecuzione sul computer oppure installare direttamente il JRE usando gli specifici gestori di pacchetti (apt-get, yum, etc) nel solo caso di Linux.

Nota: In merito alla *bitness* del JRE, in questa configurazione portatile conviene sempre installare la versione a 32 bit in modo da essere compatibile con i sistemi operativi sia a 32 bit che 64 bit

- copiare sulla chiavetta il ramo dove è stata installato il JRE, come di seguito descritto.

Nota: Il link citato può cambiare in relazione allo sviluppo delle ulteriori versioni del JRE da parte del produttore.

Facendo un esempio in ambiente **Windows** e supponendo che il JRE del nostro computer sia installato nel ramo **C:\Programmi\Java\jre7**, si dovranno copiare sulla cartella **R:\Programmi\Win_JRE** le cartelle **bin** e **lib** ed eventuali files contenuti sul computer nel ramo di installazione del JRE.

Alla fine, si dovrà ottenere una struttura analoga all'esempio riportato in figura

[Struttura JRE](#)

Struttura JRE

In ambiente **Linux**, partendo dal presupposto che gli eseguibili puntano spesso a link simbolici a livelli multipli, è necessario interrogare opportunamente il sistema per trovare la posizione sul filesystem della JRE.

L'istruzione:

```
readlink -e $(which java)
```

ci darà un output del tipo:

```
/usr/lib/jvm/java-6-sun-1.6.7.8/jre/bin/java
```

Nota: **readlink** è una utility facente parte del pacchetto **coreutils**.

Per copiare sulla chiavetta, in ambiente Linux, il JRE, usiamo il comando (senza interruzioni di riga):

```
cp -pr /usr/lib/jvm/java-6-sun-1.6.7.8/jre/* /media/  
disc/Programmi/Linux_JRE
```

Qualora non si desideri conservare il JRE sul computer, a questo punto è possibile rimuoverlo mediante il rispettivo gestore di pacchetti specifico del sistema operativo.

[[modifica](#)] Terzo passo - FidoCadJ

Questo passaggio consiste nella semplice operazione di copia del file **fidocadj.jar** nella chiavetta, in corrispondenza della directory

- su Windows:

```
R:\Programmi\FidoCadJ_portable
```

- su Linux:
/media/disc/Programmi/FidoCadJ_portable

L'operazione può essere eseguita mediante **Explorer** di **Windows** oppure da riga di comando.

[[modifica](#)] Quarto passo - gli scripts di attivazione

Il quarto ed ultimo passaggio comporta la creazione di due *shell scripts* - uno per ciascun sistema operativo - che permetteranno di mandare in esecuzione **FidoCadJ** anche su altri computer Windows XP oppure Linux

- Windows (script fidocadj.bat)

```
@echo off
set PATH=.;..\FidoCadJ_portable\Win_JRE\bin
set CLASSPATH=.;..\FidoCadJ_portable\Win_JRE\lib;..\FidoCadJ_portable
..\FidoCadJ_portable\Win_JRE\bin\java.exe -jar ..\FidoCadJ_portable\fidocadj.jar
```

- Linux (script fidocadj.sh)

```
#!/bin/bash
export PATH=../../FidoCadJ_portable/Linux_JRE/bin
export CLASSPATH=../../FidoCadJ_portable/Linux_JRE/lib:../../FidoCadJ_portable
../../FidoCadJ_portable/Linux_JRE/bin/java -jar ../../FidoCadJ_portable/fidocadj.jar
```

Assicuriamoci che i due scripts **fidocadj.bat** e **fidocadj.sh** abbiano i necessari privilegi di esecuzione nel rispettivo sistema operativo.

Idealmente, i due scripts vanno posizionati nella stessa directory che contiene **fidocadj.jar** ma possono funzionare anche in una sottodirectory specifica interna alla **Programmi**; naturalmente, modificando opportunamente gli scripts è possibile posizzarli dove si preferisce purchè siano rispettati i puntamenti alle variabili *PATH* e *CLASSPATH*, agli eseguibili del JRE *java/java.exe* ed al file *fidocadj.jar*. In caso di modifica degli scripts, l'aspetto fondamentale è di riferire i puntamenti a files e directories in modo *relativo*, in modo da conservare la portabilità dell'applicazione indipendentemente dal nome dell'unità (Windows) oppure del mount point (Linux) su cui è montata la chiavetta.

[[modifica](#)] Prova di funzionamento

Per verificare se i passi sono stati eseguiti correttamente (e se non ho trascurato nulla di importante), si può fare una prova di funzionamento, smontando la chiavetta dal computer che abbiamo usato finora e montandola su un'altro computer (Windows oppure Linux, sprovvisto di JRE) e lanciando gli scripts con le rispettive modalità di Windows oppure di Linux; l'apertura della nota finestra principale di **FidoCadJ** confermerà che abbiamo agito correttamente:

[Finestra FidoCadJ](#)

Finestra FidoCadJ

[[modifica](#)] Approfondimenti

Per chi desiderasse maggiori informazioni in merito alla modifica degli *shell scripts* ed ai *filesystems*, suggerisco di iniziare da questi link:

- [Programmazione Bash \(Linux\)](#)
- [Programmazione Batch \(Windows\)](#) (copia su Archive.org)
- [Breve guida ai filesystems](#)

Infine, volevo ringraziare [admin](#) per l'aiuto nell'impostazione grafica e per la bella foto allegata all'articolo in Home Page.

Estratto da "<https://www.electroyou.it/mediawiki/index.php?title=UsersPages:Alev:fidocadj-portabile-un-cae-al-guinzaglio>"