

## Parsing di un file Excel con Bash

Il parsing di file Excel (con estensione `.xlsx` o `.xls`) non è un compito immediato in un ambiente Bash puro, poiché questi file non sono semplici file di testo leggibili direttamente dalla shell. Tuttavia, ci sono alcuni strumenti e approcci che permettono di estrarre i dati contenuti in un file Excel e manipolarli da uno script Bash.

In questo articolo, esploreremo come effettuare il parsing di file Excel utilizzando strumenti esterni e come integrarli in un flusso di lavoro in Bash. Analizzeremo principalmente due soluzioni:

1. **Convertire il file Excel in CSV:** Utilizzare strumenti come `ssconvert` o `xlsx2csv` per convertire un file Excel in un formato più gestibile, come il CSV, e poi manipolarlo con i comandi standard di Bash.
2. **Utilizzare `in2csv` o `xlsx2csv` direttamente in Bash:** Esistono strumenti che possono trasformare il file Excel in formato CSV leggibile direttamente.

Il primo approccio consiste nel convertire il file Excel in un file CSV, che è molto più facile da manipolare in Bash. Per fare questo, possiamo usare il comando `ssconvert`, parte della suite **Gnumeric**.

Su sistemi basati su Debian/Ubuntu, puoi installare `ssconvert` con il seguente comando:

```
sudo apt-get install gnumeric
```

Su altri sistemi, il metodo di installazione potrebbe variare, ma generalmente puoi usare il gestore di pacchetti del sistema.

Supponiamo di avere un file Excel chiamato `data.xlsx`. Per convertirlo in CSV, possiamo eseguire il comando:

```
ssconvert data.xlsx data.csv
```

Questo creerà un file CSV, che è facilmente leggibile con comandi standard come `cat`, `awk`, `cut`, `grep` e così via. Ad esempio, per leggere la prima colonna del file CSV, potresti usare:

```
awk -F, '{ print $1 }' data.csv
```

Se desideri evitare di installare un pacchetto più pesante come Gnumeric, esiste un'alternativa leggera chiamata `xlsx2csv`, che è un semplice script Python per convertire file `.xlsx` in CSV.

Puoi installare `xlsx2csv` tramite `pip`:

```
pip install xlsx2csv
```

Per convertire un file Excel in CSV, usa il seguente comando:

```
xlsx2csv data.xlsx data.csv
```

Una volta ottenuto il CSV, puoi manipolarlo in Bash come nel metodo precedente. Ecco un esempio per leggere la seconda colonna:

```
awk -F, '{ print $2 }' data.csv
```

Un'altra alternativa interessante è `in2csv`, parte della suite **csvkit**, che permette di leggere direttamente i file Excel e convertirli in CSV, il tutto all'interno della pipeline Bash.

Installa `csvkit` con `pip`:

```
pip install csvkit
```

Puoi convertire e stampare il contenuto di un file Excel direttamente nel terminale con il seguente comando:

```
in2csv data.xlsx
```

Puoi anche indirizzare l'output in un file:

```
in2csv data.xlsx > output.csv
```

Un esempio completo di script Bash per convertire un file Excel in CSV e manipolarne i dati potrebbe essere:

```
#!/bin/bash

# Controlla se il file Excel esiste
if [ ! -f "$1" ]; then
```

```
    echo "File Excel non trovato!"
    exit 1
fi

# Converti il file Excel in CSV
ssconvert "$1" output.csv

# Esegui parsing del file CSV (ad esempio, stampa la prima
colonna)
awk -F, '{ print $1 }' output.csv
```

## Conclusione

Effettuare il parsing di un file Excel con Bash richiede l'utilizzo di strumenti esterni per la conversione o la lettura dei file Excel, poiché questi non sono nativamente supportati dalla shell. Utilizzando strumenti come `ssconvert`, `xlsx2csv` o `in2csv`, è possibile convertire i file Excel in CSV e quindi manipolare i dati con comandi standard di Bash. Questo approccio è versatile e permette di integrare la gestione di file Excel in un flusso di lavoro completamente basato su Bash.