

UPLOAD DI FILE IN PHP e Permessi (2021)

E' di uso comune per la applicazioni web poter "uploadare" file (ad esempio foto, filmati, testo) dal client verso il server.

Nel client selezionamo il file da spedire al server tramite un form HTML, con il tipo `<input type=file>`

il server riceve il file tramite uno script PHP e lo salva nel suo file system.

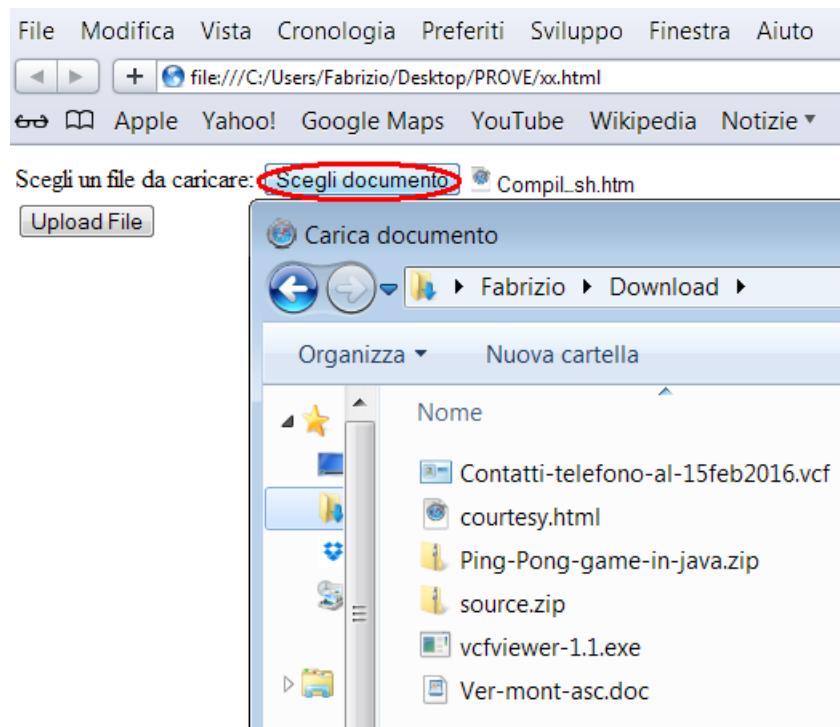
Ovviamente lo script PHP deve essere in grado di reperire il file selezionato dall'utente e di manipolarlo (ad esempio spostarlo in una determinata cartella).

E' importante anche impostare una dimensione massima di file per accettare il file in upload sul server. L'esempio sotto riporta un valore di circa 100Kb.

Esempio di modulo HTML:

```
<form enctype="multipart/form-data" action="uploader.php"
method="POST">
<input type="hidden" name="MAX_FILE_SIZE" value="100000">
Scegli un file da caricare:
<input name="fileupload" type="file"><br>
<input type="submit" value="Upload File">
</form>
```

Come si vede nel browser:



Programma PHP lato server (uploader.php):

▼ All'esecuzione di uploader.php il file esiste già ed è in una locazione temporanea nel server. Occorre spostare il file se non vogliamo perderlo!

▼ I file caricati vengono memorizzati nell'array globale \$_FILES.

Codice PHP per l'upload:

```
<?php

$file_dir = "/home/dsacco/www/temp/";
//Cartella dove scaricare... CAMBIALA PURE...

$file=$_FILES['fileupload'];
// $file= Vettore contenente informazioni sul file caricato....

printf("<hr>Nomefile=%s<hr>dimensione %s bytes<hr>salvato in %s<hr>",
       $file['name'], $file['size'], $file['tmp_name']);
// visualizzo informazioni del file

if (is_uploaded_file($file['tmp_name']))
// Se il file è stato caricato sul server...
{
    move_uploaded_file($file['tmp_name'], $file_dir.$file['name'])
    or die ("Errore nello spostare il file caricato...");
    printf( "Il file è stato spostato nella directory: %s!<hr>",
           $file_dir);
    chmod($file_dir.$file['name'],0777);
    // Superfluo, ma potrebbe essere utile; attenzione ai permessi
}
?>
```

Approfondimento per UPLOAD su server unix/linux: PERMESSI e comando chmod

I permessi non sono altro che le **autorizzazioni necessarie affinché un utente possa accedere ai file ed alle cartelle del sistema**. In altre parole: ciascun utente può accedere o meno ad un determinato file o cartella a seconda che l'amministratore di sistema gli abbia conferito o meno la necessaria autorizzazione.

I permessi possono consentire ad un utente (o gruppo di utenti) di accedere ad un file o una cartella in lettura (r), scrittura (w) o esecuzione(x). Vediamo nel dettaglio cosa significa:

lettura (r come read) - l'utente (o il gruppo cui appartiene) puoi leggere il contenuto; nel caso delle directory l'utente potrà vedere i file e le sotto-cartelle in quest'ultima contenute;

scrittura (w come write) - l'utente (o il gruppo cui appartiene) può modificare il contenuto di un file; nel caso delle directory permette di modificarne il contenuto aggiungendo o rimuovendo elementi al suo interno;

esecuzione (x come execute) - l'utente (o il gruppo cui appartiene) può eseguire un file eseguibile; nel caso delle directory consente di accedere al loro contenuto.

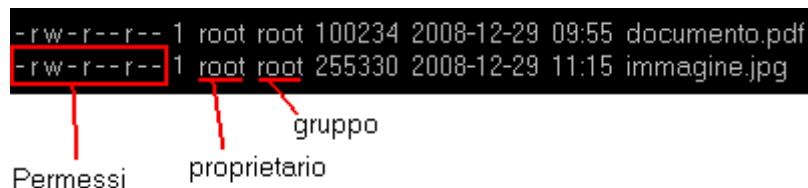
Cartelle, file e permessi

In diverse lezioni della nostra guida ci siamo soffermati sulle modalità di gestione di file e cartelle. In questa lezione vedremo un argomento strettamente correlato: la **gestione dei permessi** (argomento molto importante essendo Linux un sistema *multi-utente* come detto nella lezione precedente e ricordato poco sopra).

Abbiamo [già visto](#) il comando **ls**, in particolar modo abbiamo già detto che l'utilizzo della sintassi:

```
ls -al
```

ha la funzione di stampare a video l'elenco dei file (anche quelli nascosti) e delle cartelle presenti nella posizione corrente. L'output generato da questo comando è ricco di informazioni:



```
-rw-r--r-- 1 root root 100234 2008-12-29 09:55 documento.pdf
-rw-r--r-- 1 root root 255330 2008-12-29 11:15 immagine.jpg
```

Permessi proprietario gruppo

Nell'immagine qui sopra si evidenziano le informazioni utili ai fini di questa lezione:

colonna dei permessi: UGO (utente / proprietario / gruppo)

La **colonna dei permessi** contiene 10 lettere (o trattini):

(il **primo spazio** indica la tipologia dell'elemento e può avere i seguenti valori: **d** (directory), **l** (link simbolico), **-** (file))

i nove caratteri successivi indicano i permessi. Più precisamente si tratta di tre distinti gruppi di 3 permessi (**r** = lettura; **w** = scrittura; **x** = esecuzione, **-** = non permesso). Il primo gruppo da tre riguarda il *proprietario*, il secondo riguarda il *gruppo* ed il terzo riguarda gli *altri utenti*.

Nel nostro esempio si tratta di due file, per entrambi il proprietario può leggere e scrivere (rw-), mentre il gruppo e gli altri utenti possono solo leggere (r--).

SCHEMA BINARIO PER chmod

Esempio:

se voglio **rw-rw-r--** (completo per proprietario, Lettura/Scrittura per il gruppo e sola Lettura per gli altri)

ricavo il binario: 111110100 → passo ad ottale -->764 → lancio chmod → chmod 764 nomefile