

BigFile

Listet die Verzeichnis Struktur rekursiv und zeigt die Verzeichnisgröße bzw. auch Dateien welche größer sind als eine angegebene Größe

Bei Gelegenheit kann man noch folgende Unix Befehle mit einbinden

df

disk free zeigt den freien, also verfügbaren, Speicherplatz auf der Festplatte an.

Freien Speicherplatz des gesamten Dateisystems anzeigen:

```
df -h
```

Freien Speicherplatz des lokalen Dateisystems anzeigen:

```
df -hl
```

du

disk usage zeigt den belegten, nicht verfügbaren, Speicherplatz auf der Festplatte an.

Gesamten belegten Speicherplatz des Systems anzeigen:

```
du -sh /
```

Belegten Speicherplatz des Ordners Media anzeigen, ohne *.bmp Dateien zu berücksichtigen:

```
du --exclude="*.bmp*" -sh media/
```

Belegten Speicherplatz des aktuellen Verzeichnisses anzeigen:

```
du -sh ./
```

Sortierung des belegten Speicherplatz mit maximaler Rekursions-Tiefe von 1:

```
du -h --max-depth 1 ./ | sort -h
```

[bigfile.php](#)

```
<?php
/*
 * Listet die Verzeichnis Struktur rekursiv und zeigt die
 * Verzeichnisgröße
```

```
* bzw. auch Dateien welche größer sind als eine angegebene Größe
* Copyright by Conny Henn 2017 - www.Hennweb.de
*/

// Einstellungen
$folder = "/var/customers/webs/No1"; // Verzeichnis ab dem rekursiv
angezeigt werden soll | aktuelles Verzeichnis = dirname(__FILE__);
$big_file = True; // Zeige Dateien an die größer sind
als $big_file_size, TRUE = ja, FALSE = nein
$big_file_size = 1000000; // Zeigt Dateien an die größer
sind als 1000000 Bytes = 1 MB
$big_dir_size = 1000000; // kennzeichnet Verzeichnisse
deren Inhalt größer als 1000000 Bytes = 1 MB in rot
$empty_dir = false; // Leere Verzeichnisse anzeigen TRUE oder
nicht FALSE

// ab hier nichts mehr ändern
echo "<pre>Webespace verbrauchter Platz (C) by Conny Henn<br>";
echo $folder."<br>";
echo dirname(__FILE__)."<br>";
echo "Gesamtgröße: ".
FileSizeConvert(disk_total_space("/"))."<br>";
echo "Frei : ". FileSizeConvert(disk_free_space("/"))."<br>";

$size = 0;
$dir_size = 0;
$dir = new RecursiveIteratorIterator(new
RecursiveDirectoryIterator($folder),
RecursiveIteratorIterator::CHILD_FIRST);
foreach($dir as $file)
{
    if ( is_dir($file->getPathname()) )
    {
        if (substr($file->getPathname(),-1,1) <> ".")
        {
            if ($empty_dir){
                $dir_size_value=0;
            } else {
                $dir_size_value=1;
            }

            // recursive Verzeichniss Größe -Verzeichnis mit
            Unterverzeichnis
            // $dir_size = GetDirectorySize( $file->getPathname()
        );
    }
}
```

```

// Verzeichnissgröße - NUR aktuelles Verzeichnis
$dir_size = dirSize( $file->getPathname() );

if ( $dir_size >= $dir_size_value )
{
    if ( $dir_size > $big_dir_size ) {
        echo "<font color='red'>";
        echo substr("
".FileSizeConvert($dir_size), -10).' '. $file->getPathname().'<br>';

        echo "</font>";
        //if ( $dir_size > $big_dir_size )
    }
    $dir_size=0;
}
}
} else {
    $x = filesize($file->getPathname());
    $size += $x;
    $dir_size += $x;
    if ($x>$big_file_size && $big_file)
    {
        echo '<font color="blue">FILE      >
'. $file->getPathname(). " - ".FileSizeConvert($x).'</font><br>';
    }
}
}
echo "Gesamtverbrauch: ".FileSizeConvert($size);

// -----
function FileSizeConvert($bytes)
{
    $units = array(' B','KB','MB','GB','TB');
    $bytes = max($bytes, 0);
    $pow = floor(($bytes ? log($bytes) : 0) / log(1024));
    $pow = min($pow, count($units) - 1);
    $bytes /= pow(1024, $pow);
    return round($bytes, $precision) . ' ' . $units[$pow];
    return $result;
}

// recursive Verzeichniss Größe - Verzeichnis mit Unterverzeichnis
function GetDirectorySize($path){
    $bytestotal = 0;
    $path = realpath($path);
    if($path!==false && $path!='' && file_exists($path)){
        foreach(new RecursiveIteratorIterator(new
RecursiveDirectoryIterator($path, FilesystemIterator::SKIP_DOTS)) as
$object){

```

```
        $bytestotal += $object->getSize();
    }
}
return $bytestotal;
}

// Verzeichniss Größe - NUR aktuelles Verzeichnis
function dirSize($directory) {
    $size = 0;
    foreach(new RecursiveIteratorIterator(new
RecursiveDirectoryIterator($directory)) as $file){
        $size+=$file->getSize();
    }
    return $size;
}

?>
```

From:
<https://wiki.hennweb.de/> - **HennWeb**

Permanent link:
<https://wiki.hennweb.de/doku.php?id=programmieren:php:programme:bigfile&rev=1662371265>

Last update: **05/09/2022 10:47**

