

How to Use PHP and Regex to Scan Large Files for Keyword Matches?

1081 Administrator Tue, Aug 1, 2023 [Web Development](#) 0 3344

Here's a very simple PHP script to scan a text file for occurrences of a given keyword using regular expressions.

```
#!/usr/bin/php
$keyword = "your_keyword";
$file = "your_file.txt";
$count = 0;
$contents = file_get_contents($file);
if ($contents) {
    $matches = preg_match_all('/\b$keyword\b/i', $contents, $matches);
    if ($matches) {
        $count = count($matches[0]);
    }
}
echo "The keyword '$keyword' was found $count times in the file '$file'.";
```

In the script above, we're using `preg_match_all` function, which performs a global regular expression match. The pattern `'/$keyword/i'` is a case-insensitive (`'i'`) match for the keyword. This script will find all occurrences of the keyword in the file and print the total number of matches.

Please note, you will need to replace "`your_keyword`" with the keyword you want to search for and "`your_file.txt`" with the path to the text file you wish to scan.

Also, this script won't be super fast on very large files, as it reads the entire file into memory before scanning. For very large files, you may want to use a more efficient method of scanning the file in chunks, such as using the `fread` function in a loop, but this would make the script more complex.

Using `fread` to search large files in chunks

Below is an example of using `fread` in a loop to scan the contents of a large file in chunks. This script reads a chunk of the file, scans it for the keyword, then moves on to

the next chunk. This allows it to handle very large files that wouldn't fit into memory.

The chunk size of `8192` bytes (8KB) is just an example. You may adjust the chunk size based on the available memory and the file size you are dealing with. Please replace "your_keyword" with the keyword you want to search for and "**your_file.txt**" with the path to the text file you want to scan.

Be aware that if your keyword splits across two chunks, this method will fail to detect it. For files where this is a potential issue, you would need a more complex solution to handle that scenario.

Online URL:

<https://www.articlediary.com/article/how-to-use-php-and-regex-to-scan-large-files-for-keyword-matches-1081.html>

SyntaxHighlighter.config.stripBrs=true; SyntaxHighlighter.all();