strpbrk

Defined in header `<string.h>`

```
char* strpbrk( const char* dest, const char* breakset );
```

Scans the null-terminated byte string pointed to by `dest` for any character from the null-terminated byte string pointed to by `breakset`, and returns a pointer to that character.

The behavior is undefined if either `dest` or `breakset` is not a pointer to a null-terminated byte string.

Parameters

dest	-	pointer to the null-terminated byte string to be analyzed
breakset	-	pointer to the null-terminated byte string that contains the characters to search for

Return value

Pointer to the first character in `dest`, that is also in `breakset`, or null pointer if no such character exists.

Notes

The name stands for "string pointer break", because it returns a pointer to the first of the separator ("break") characters.

Example

Run this code

```
#include <stdio.h>
#include <string.h>

int main(void)
{
    const char* str = "hello world, friend of mine!";
    const char* sep = " ,!";

    unsigned int cnt = 0;
    do {
       str = strpbrk(str, sep); // find separator
       if(str) str += strspn(str, sep); // skip separator
       ++cnt; // increment word count
    } while(str && *str);

    printf("There are %u words\n", cnt);
}
```

Output:

```
There are 5 words
```

References

- C11 standard (ISO/IEC 9899:2011):
  - 7.24.5.4 The strpbrk function (p: 368)
- C99 standard (ISO/IEC 9899:1999):
  - 7.21.5.4 The strpbrk function (p: 331)
- C89/C90 standard (ISO/IEC 9899:1990):
  - 4.11.5.4 The strpbrk function

See also

strcspn	returns the length of the maximum initial segment that consists of only the characters not found in another byte string (function)
strchr	finds the first occurrence of a character (function)
strtok **strtok_s** (C11)	finds the next token in a byte string (function)

C++ documentation for **strpbrk**

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=c/string/byte/strpbrk&oldid=96278"