

strtok "string tokenize"

char\* strtok(char\* str, char\* delimiters)

Repeatedly split str on the given delimiters and return the start of the next token

```
int main() {
    char cmd[] = "cp pa2.c pa2-backup.c";
```

```
    char* first = strtok(cmd, " ");
```

```
    char* second = strtok(NULL, " ");
```

```
    char* third = strtok(NULL, " ");
```

```
    char* fourth = strtok(NULL, " ");
```

```
    char* args[] = {first, second, third};
```

Variable/Role

Address

Data

0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F

0x...00

0x...08

0x...10

0x...18

0x...20

0x...28

0x...30

0x...38

0x...40

0x...48

0x...50

0x...58

0x...60

0x...68

0x...70

0x...78

0x...80

0x...88

0x...90

0x...98

0x...A0

0x...A8

0x...B0

0x...B8

0x...C0

0x...C8

0x...D0

0x...D8

0x...E0

0x...E8

0x...F0

0x...F8

cmd

first

second

third

fourth

Diagram illustrating the string "cp pa2.c pa2-backup.c" with tokens and delimiters marked. The string is shown in memory, with characters 'c', 'p', ' ', 'p', 'a', '2', '.', 'c', ' ', 'p', 'a', '2', '-', 'b', 'a', 'c', 'k', 'u', 'p', '.', 'c'. The first token "cp" is marked with a green star. The second token "pa2.c" is marked with an orange square. The third token "pa2-backup.c" is marked with a red triangle. The string ends with a null terminator '\0'. A red triangle points to the null terminator with the text "found end of string!".

0x60

0x63

0x69

NULL