

# Sistemas Operativos

## 1. Tokenize

Lea la documentación de la función de librería `tokenize(3)` en la página <https://swtch.com/plan9port/man/man3/getfields.html> e implemente una similar en C en Linux pero sin escapado (*quoting*) y que funcione sólo para strings normales, es decir, arrays de char acabados en `'\0'`. La función a implementar,

```
int mytokenize(char *str, char **args, int maxargs);
```

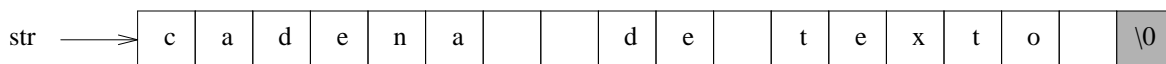
recibirá una cadena de caracteres `str`, un array de punteros `args` y como tercer parámetro, `maxargs` el número máximo de entradas en `args`.

La cadena `str` se separará en subcadenas cuando se encuentren tabuladores `'\t'`, retornos de carro `'\r'`, espacios `' '` o fines de línea `'\n'`, que llamaremos caracteres separadores. Las subcadenas se separarán sobrescribiendo el primer carácter separador que delimite la cadena con un `'\0'` o carácter nulo. Cada puntero en el array `args` deberá apuntar al principio la subcadena de `str` correspondiente.

El valor de retorno será el número de subcadenas encontradas, menor o igual que `maxargs`. Por ejemplo tras llamar a

```
mytokenize(str, args, 3);
```

con la cadena de la Figura 1,

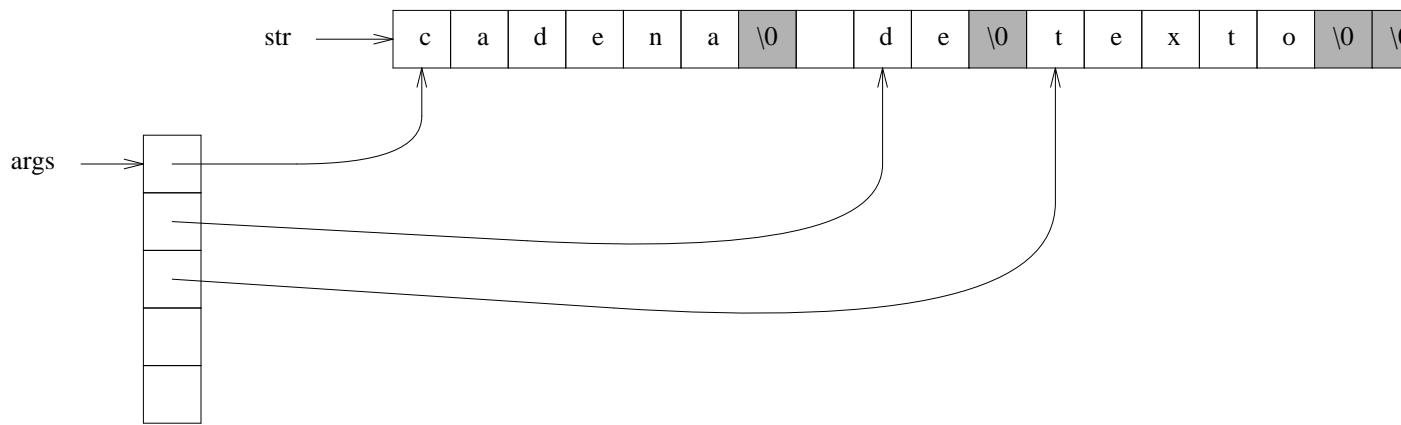


**Figura 1.**

tendremos el resultado que se ve en la Figura 2.

Los punteros `args[3]` y `args[4]` conservarán en este caso el valor que tuviesen antes de la llamada. El valor de retorno es 3.

N caracteres separadores seguidos no separan una subcadena vacía, sino que se considerarán todos juntos como el mismo delimitador. Esto se puede ver en el ejemplo al final de la subcadena apuntada por `args[0]`.



**Figura 2.**