

# Boletín 4: El comando `tr`

## Ampliación de Sistemas Operativos

Dpto. Ingeniería y Tecnología de Computadores (DITEC)

Universidad de Murcia

Curso 2017/2018

## 1 El comando `trod`

# 1. El comando `trod`

- (2.0p) **EJERCICIO 1:** Implementa el comando interno trod:

```
1  simplesh> echo a1b2c3d4 | trod abcd ABCD
    A1B2C3D4
```

- Para ello, únicamente se pueden usar llamadas al sistema POSIX o funciones de la librería estándar de C (C11)
- Llamadas POSIX y/o glibc a considerar:
  - getopt()
  - read()
  - write()
  - strchr()
  - fsync()
- Para implementar trod, escribe la función run\_trod() e inserta llamadas a la misma en simplesh.c donde sea necesario

# El comando trod

- Si se proporciona el parámetro `-h`, trod debe enviar a stdout:

```
1 simplesh> trod -h
2 Uso: trod [-t TAMAÑO] [-d] [-c] SET1 [SET2]
   SET1 y SET2 son conjuntos de caracteres.
4   Opciones:
   -t TAMAÑO en bytes de los bloques leídos de 'stdin'
6   -d Borra los caracteres de SET1
   -c Comprime los caracteres de SET1
8   -h help
```

- Si se teclea sólo trod:

```
1 simplesh> trod osa OSA
2 "aso"
   "ASO"
4 [CTRL+D]
simplesh>
```

- Se deben especificar dos, y sólo dos, conjuntos de caracteres que siempre deben tener el mismo número de caracteres:

```
1 simplesh> echo a1b2c3d4 | trod abcd
run_trod: Se debe especificar tanto SET1 como SET2
3 simplesh> echo a1b2c3d4 | trod abcd ABCD 1234
run_trod: Se debe especificar tanto SET1 como SET2
5 simplesh> echo a1b2c3d4 | trod abcd ABC
run_trod: SET1 y SET2 deben tener el mismo tamaño
```

# El comando trod

- El parámetro `-t TAMAÑO` establece el tamaño en bytes de los bloques leídos de `stdin` y, por tanto, el tamaño máximo de los bloques escritos a `stdout` ( $1 \leq \text{TAMAÑO} \leq 1 \text{ MB}$ ):

```
simplesh> echo a1b2c3d4 | trod -t 2 abcd ABCD
2  A1B2C3D4
```

- Si se especifica un tamaño fuera del rango anterior, `trod` muestra un mensaje de error:

```
simplesh> echo a1b2c3d4 | trod -t 0 abcd ABCD
2  run_trod: Tamaño no válido
```

- El parámetro `-d` borra los caracteres de SET1:

```
simplesh> echo a1b2c3d4 | trod -d abcd
2  1234
```

- Si se especifica también SET2, `trod` muestra un mensaje de error:

```
simplesh> echo a1b2c3d4 | trod -d abcd ABCD
2  run_trod: Se debe especificar sólo SET1
```

- Implementación del comando trod:

- 1 Procesar los parámetros con `getopt()` (`man 3 getopt`)
  - **Nota:** Antes de volver a usar `getopt()`, se debe inicializar `optind` a 1
- 2 Leer un BLOQUE de tamaño TAMAÑO de la entrada estándar con `read()` hasta que `read()` devuelva 0 notificando el fin del fichero (EOF)
- 3 Traducir los caracteres de SET1 a SET2 o borrar los caracteres de SET1 del bloque BLOQUE leído
  - La función `strchr()` comprueba si un carácter aparece en una cadena
- 4 Escribir el bloque BLOQUE resultante en la salida estándar
- 5 Asegurar que el fichero ha sido escrito a disco con `fsync()`
- 6 La implementación de trod debe ser robusta tanto con ficheros de tamaño *ilimitado* como con lecturas/escrituras parciales
- 7 Se debe verificar que el parámetro `-t` funciona correctamente tanto con el parámetro `-d` como con el parámetro `-c`

- (0.5p) **Opcional:** El parámetro `-c` comprime los caracteres de SET1

```
simplesh> echo aaabbbccccddddddcccbbaaa | trod -c abcd
2  abcdcba
```

- Si se especifica también SET2, trod muestra un mensaje de error:

```
simplesh> echo aaabbbccccddddddcccbbaaa | trod -c abcd ABCD
2  run_trod: Se debe especificar sólo SET1
```