PEQUEÑA MEMORIA CON DATOS ENCONTRADOS Y ERRORES COMETIDOS

ÁLVARO LAMADRID AGUDO

Algunas de las cosas que he buscado para poder completar el código es **como leer todos los datos que se le pasan a filtro**, como **comprobar si existe una variable de entorno** y si tienen o no un valor y el **uso de stdout y stdin**.

Algunos de los **errores** que me han surgido son los siguientes:

- A la hora de **recorrer los datos que se le pasan al filtro** lo hacía sin inicializar el 'check', sino simplemente de la siguiente manera:

```
while( scanf() != 0) {
    if(numero>(-limite) &&  numero<limite){
        // al estar dentro del limite lo mandamos a stdout
        fprintf(stdout,"%d\n", numero);
    }else{
        // al estar fuera del limite lo mandamos a stderr
        fprintf(stderr,"%d\n", numero);
    }
    scanf("%d",&numero); // escaneamos siguiente numero
}
```

De esta manera me daba el siguiente error en la terminal:

```
iamlamadrid@LAPTOP-1C3R0RPT:/mnt/c/users/lamad/onedrive/escritorio/uc/4º año/sistemas operativos/practicas/practica1$ gcc -o filtro filtro.c
filtro.c: In function 'main':
filtro.c:42:12: error: too few arguments to function 'scanf'
    while( scanf() != 0) {
           ^~~~~
In file included from /usr/include/features.h:424:0,
                 from /usr/include/x86_64-linux-gnu/bits/libc-header-start.h:33,
                 from /usr/include/stdlib.h:25,
                 from filtro.c:1:
/usr/include/stdio.h:398:12: note: declared here
 extern int __REDIRECT (scanf, (const char *__restrict __format, ...),
```

- También a la hora de **hacer el makefile** me daban errores a la hora de poner etiqueitas, ya que lo tomaba como comando y no lo encontraba:
  Este era mi makefile con etiquetas:

```
CC = gcc

generador :
    $(CC) -o generador generador.c

filtro :
    $(CC) -o filtro filtro.c
```

Por lo que la terminal me decía que eran comandos que no encontraba.

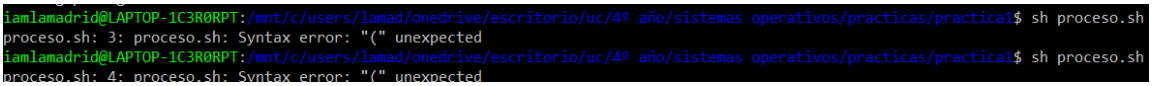
```
iamlamadrid@LAPTOP-1C3R0RPT:/mnt/c/users/lamad/onedrive/escritorio/uc/4º año/sistemas operativos/practicas/practica1$ ./makefile
./makefile: line 3: generador: command not found
./makefile: line 4: CC: command not found
./makefile: line 4: -o: command not found
./makefile: line 6: filtro: command not found
./makefile: line 7: CC: command not found
./makefile: line 7: -o: command not found
```

Así que lo he dejado como dos 'gcc' para que compile los dos programas y un par de 'echos' para ver que no ha habido fallo.

- El último fallo 'grave 'que he tenido fue el **entubar los datos a la hora de hacer el filtro** dentro de proceso.sh. Al principio me sacaba los mismos datos a los archivos stdout y stderr. Al final lo he dejado de la siguiente manera que es correcta:

```
./filtro 1000 < datos$i 2>> filtrado.stderr >> salida.stdout
```

- Por último la parte de generar los 10 números más altos de salida.stdout no lo he podido hacer porque en mi Shell no me deja usar arrays ya que me sale el error siguiente al intentar crear uno, ya que he buscado en varias páginas y todas crean de una manera los arrays y a mí no me deja, me da el error:



Por lo tanto, aporto un posible código que podría hacer esta parte:

```
array=(0 0 0 0 0 0 0 0 0 0) # array con 10 elementos a 0
controlador=0 # variables que nos sirve para ir cambiando los datos
for line in $(cat salida.stdout) # recorremos todos los datos de salida.stdout
do
    for item in ${array[*]} # recorremos el array por cada valor de salida.stdout
    do
            # si es mayor lo cambiamos y nos quedamos con el que estaba en este indice
    if[$(line) > ${array[item]}]
    then
            controlador=array[item]
            array[item]=line
            line=controlador
    fi
    done
done
# recorremos todo el array y lo vamos mostrando
for j in ${array[*]}
do
    printf "   %s\n" $item
done
```