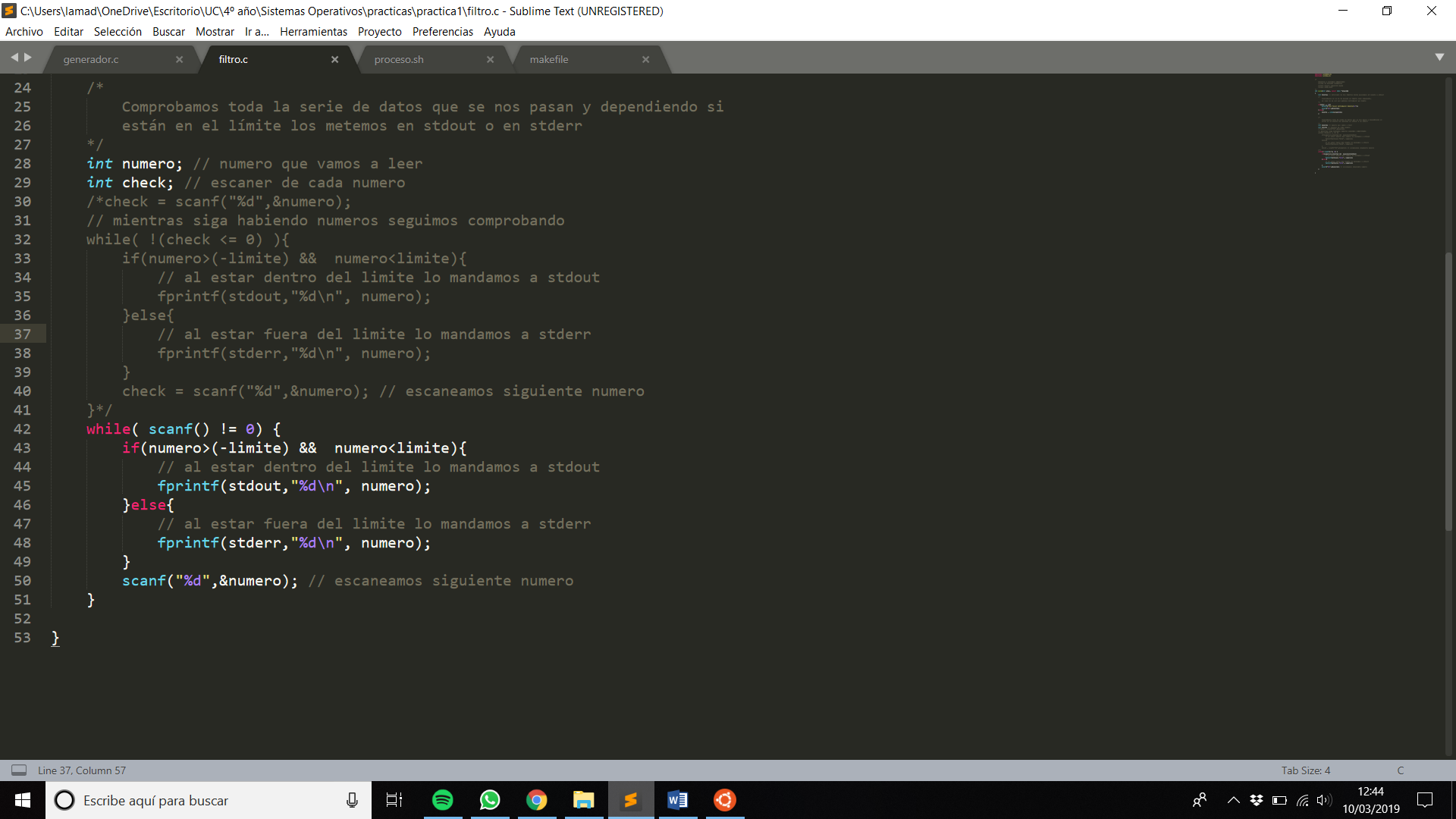
## PEQUEÑA MEMORIA CON DATOS ENCONTRADOS Y ERRORES COMETIDOS

## ÁLVARO LAMADRID AGUDO

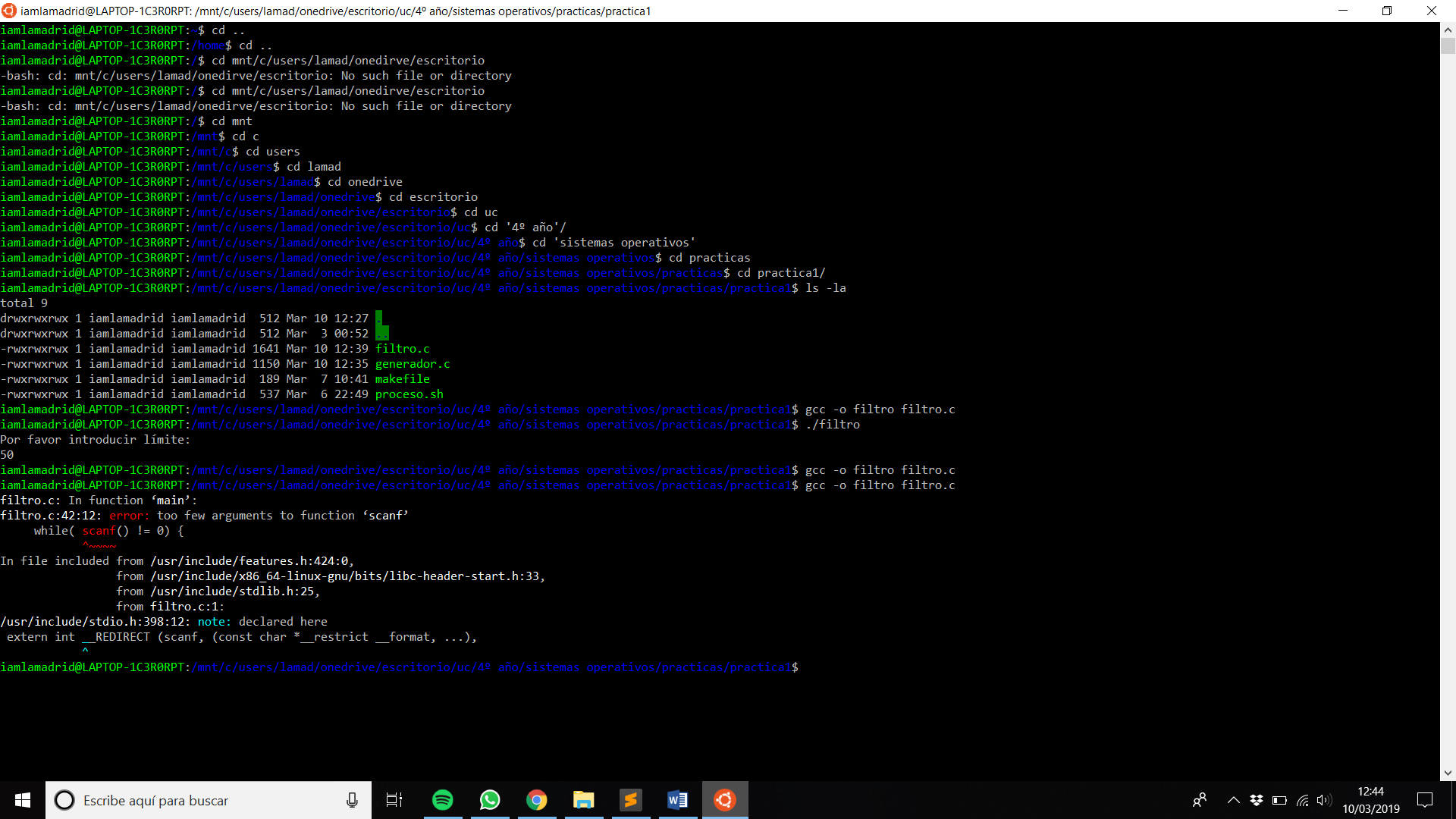
Algunas de las cosas que he buscado para poder completar el código es **como leer todos los datos que se le pasan a filtro**, como **comprobar si existe una variable de entorno** y si tienen o no un valor y el **uso de stdout y stdin**.

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:

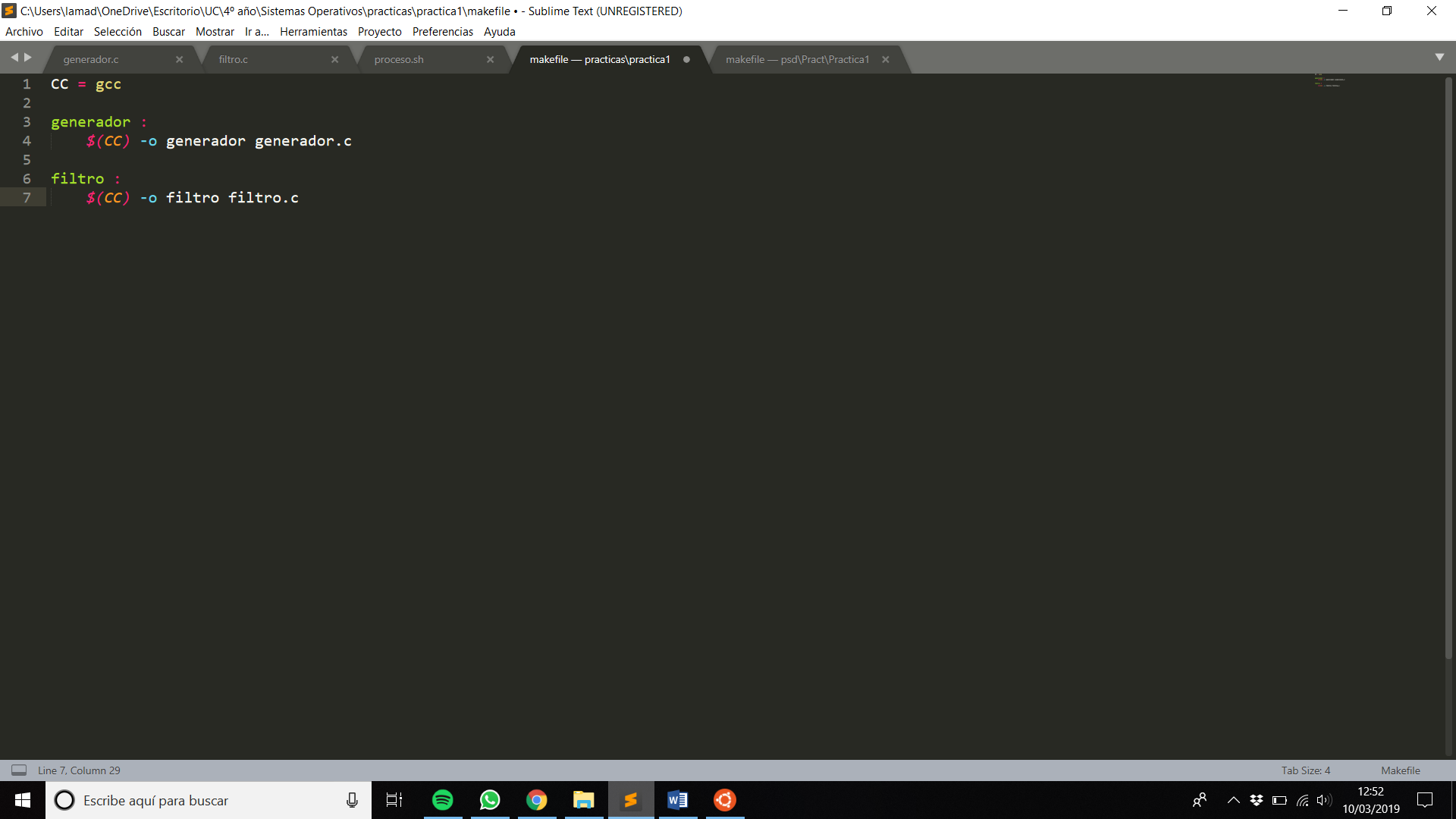


De esta manera me daba el siguiente error en la terminal:

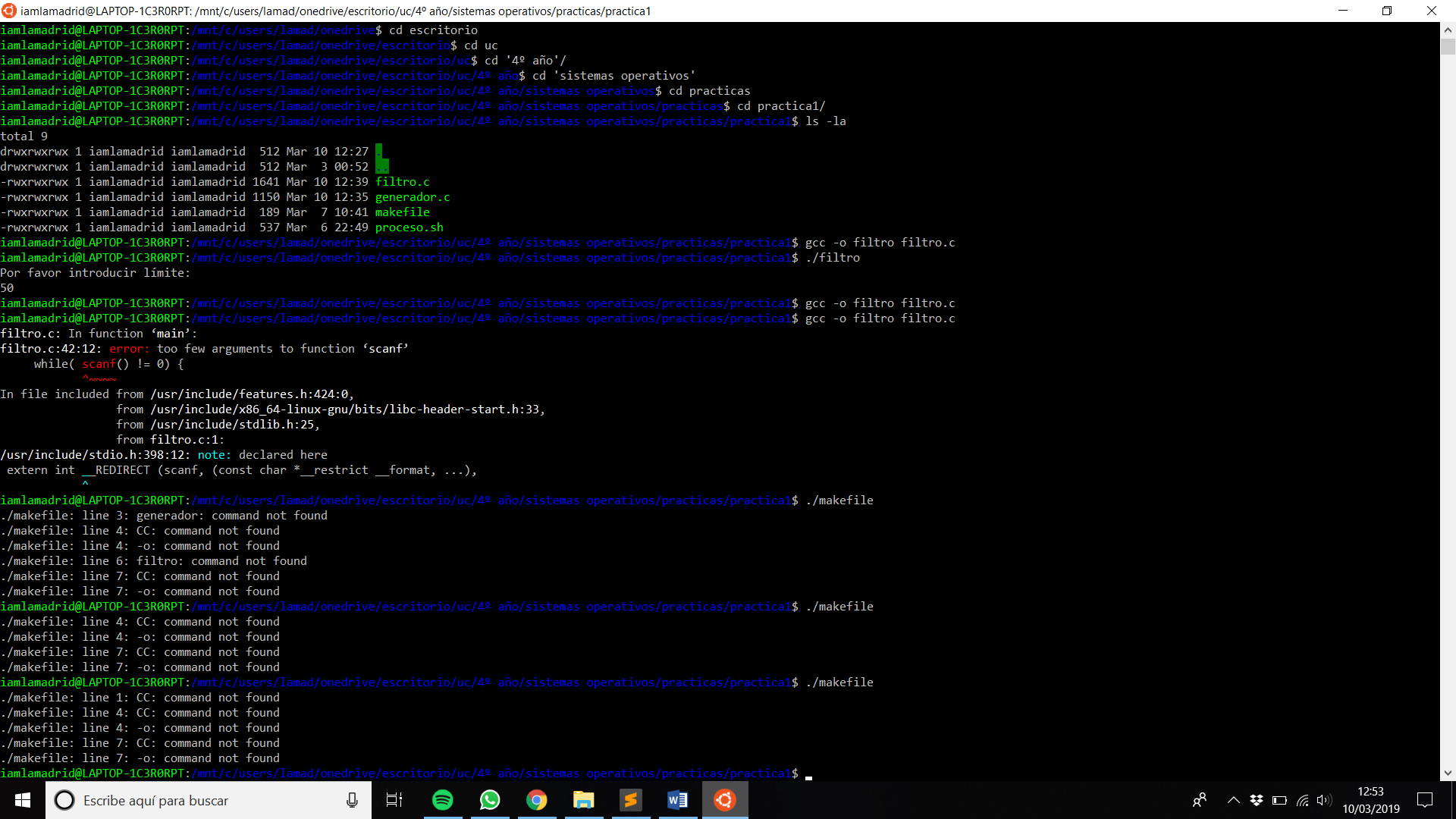


* También a la hora de **hacer el makefile** me daban errores a la hora de poner etiqueitas, ya que lo tomaba como comando y no lo encontraba:

Este era mi makefile con etiquetas:

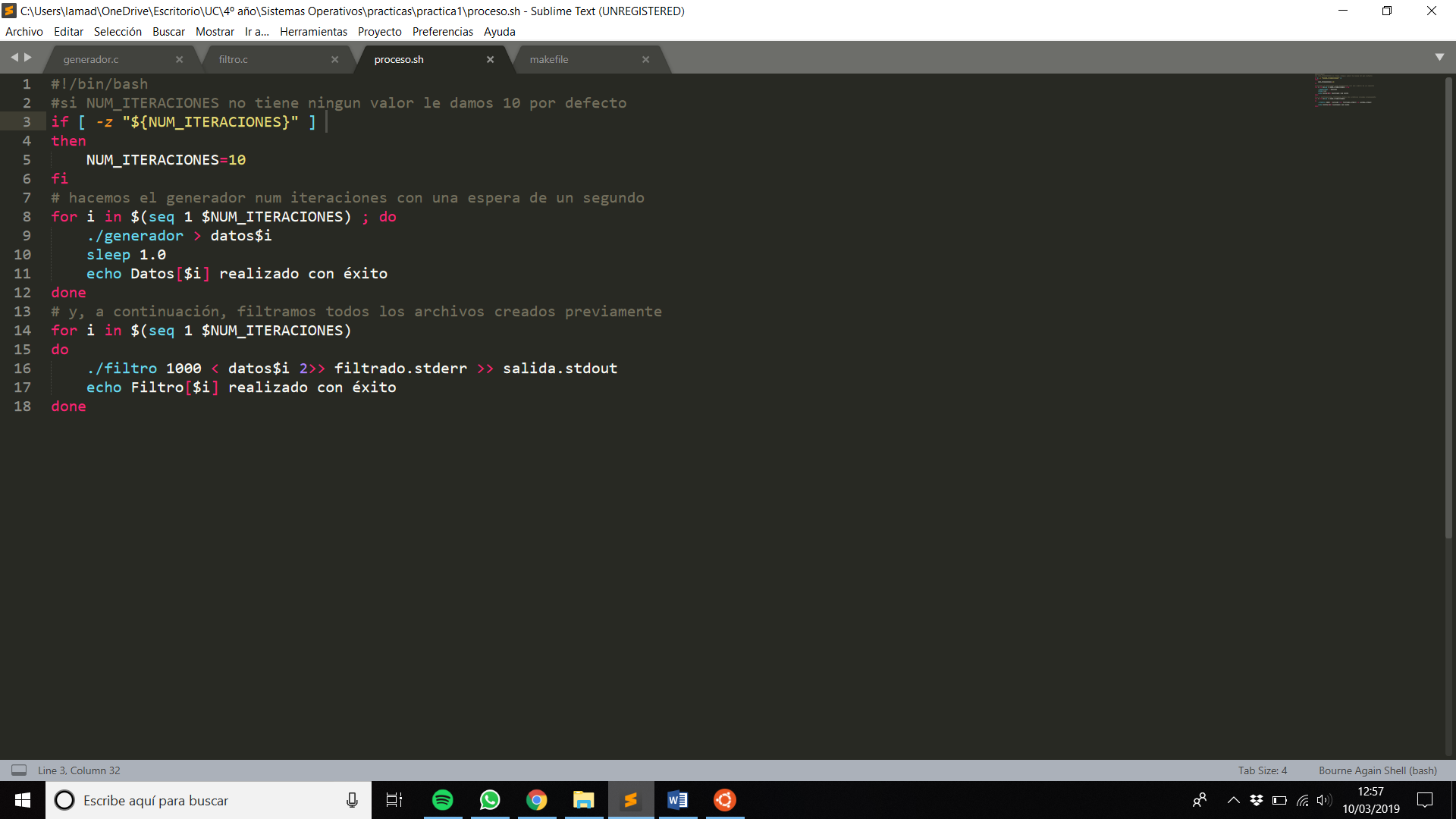


Por lo que la terminal me decía que eran comandos que no encontraba.

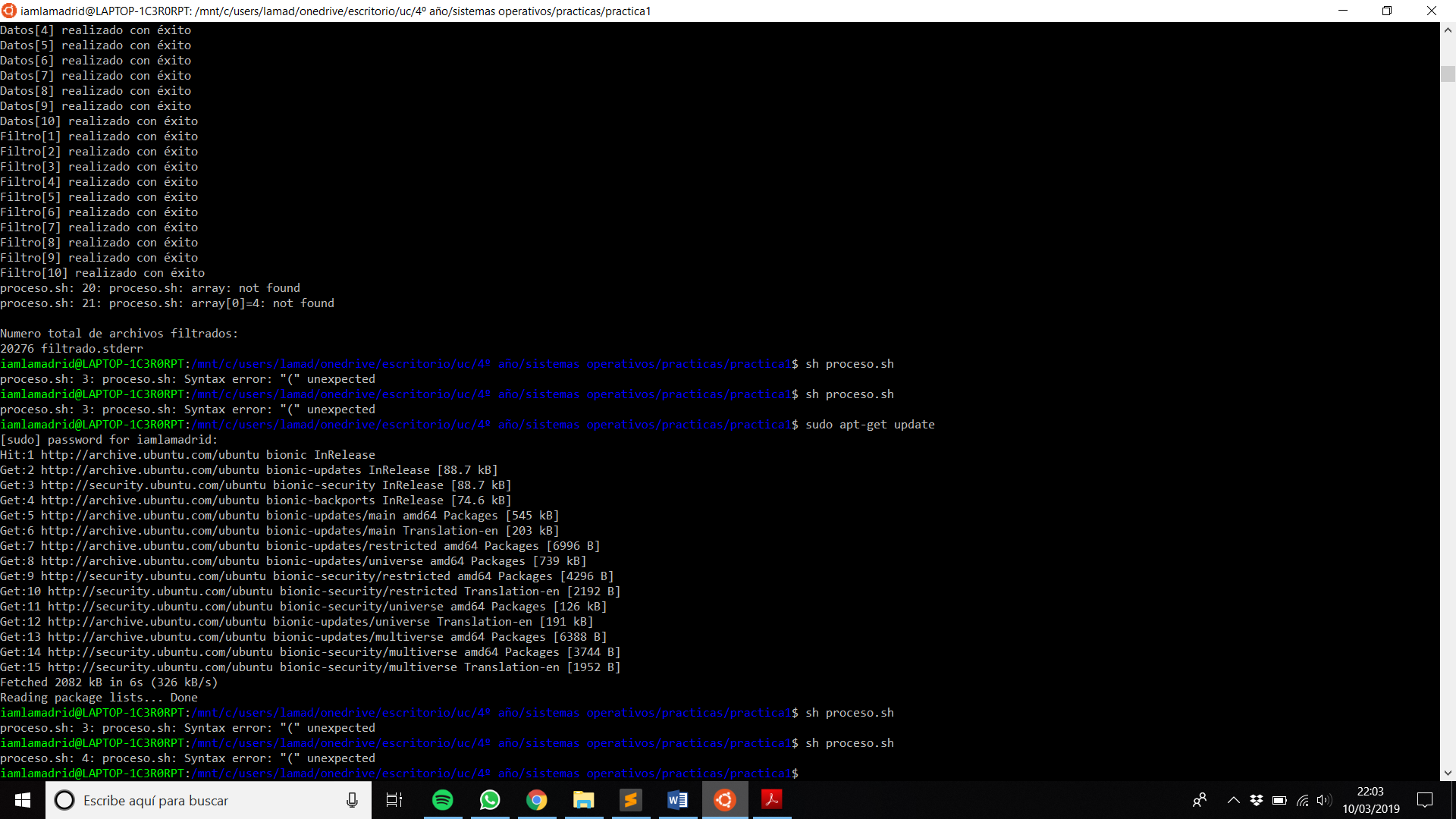


Así que lo he dejado como dos ‘gcc’ para que compile los dos programas y un par de ‘echos’ para ver que no ha habido fallo.

* 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:



* 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