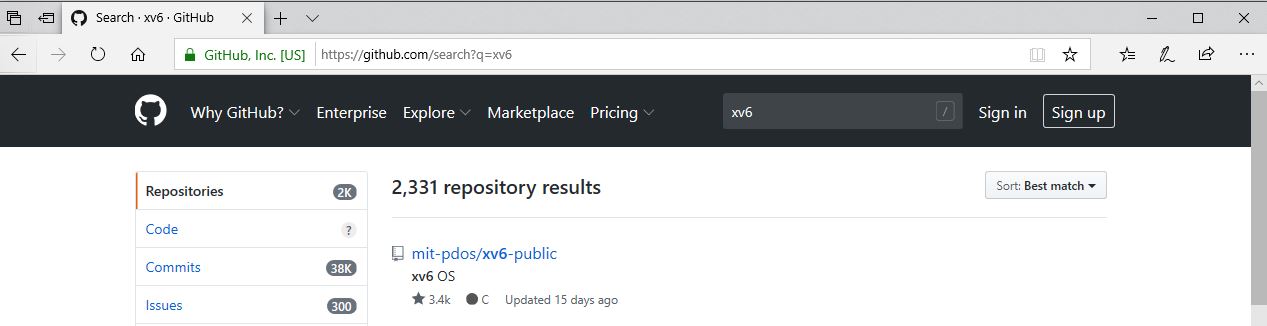
Practica 00 Construcción y ejecución del Sistema Operativo xv6

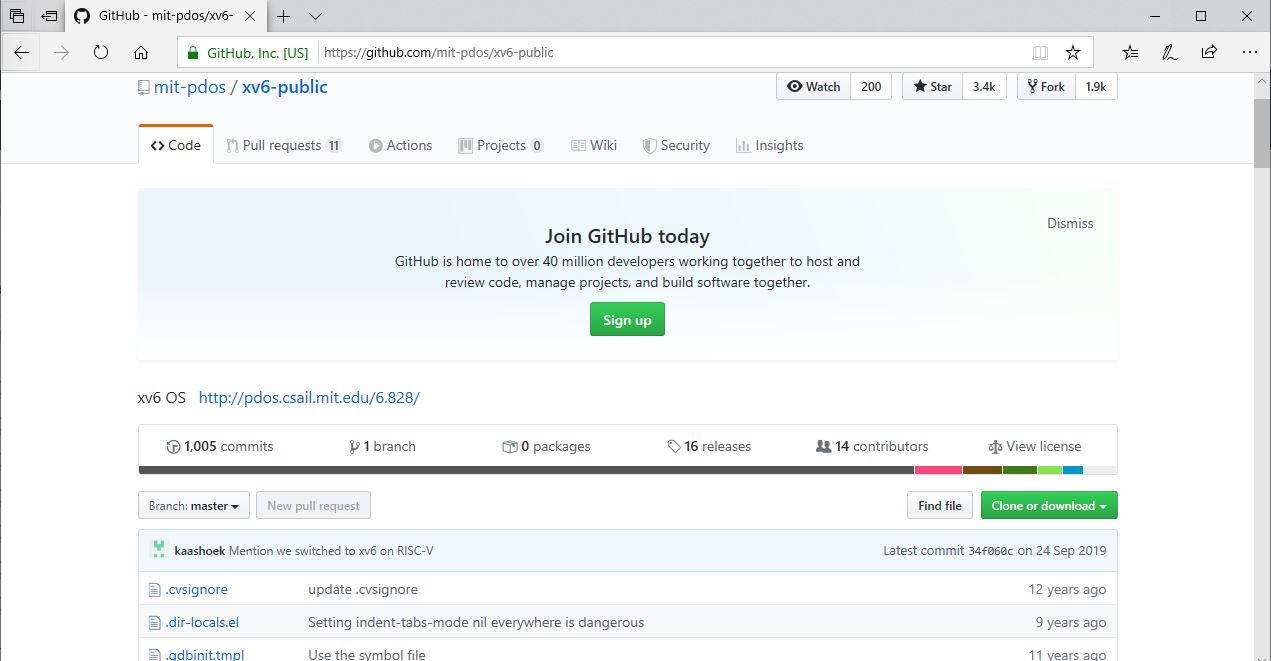
Descargar el c\'{o}digo fuente del sistema operativo xv6

Buscar las fuentes de xv6 en github.com



Resultados de la búsqueda de xv6 en github.com

Si damos clic en el repositorio xv6-public de la organización mit-pdos llegaremos al repositorio de xv6:

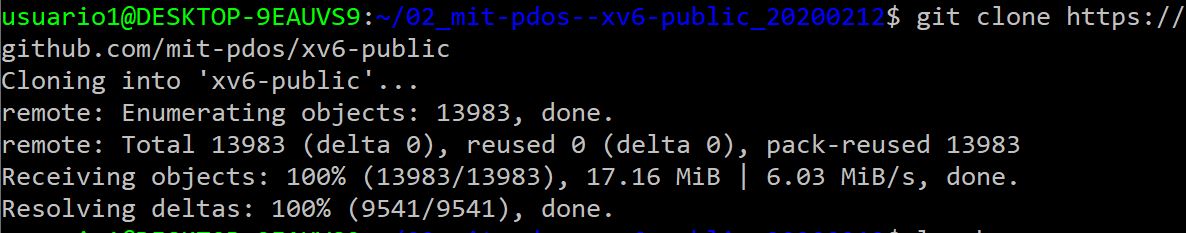


Repositorio xv6-public de la organización mit-pdos

Desde Debian GNU-Linux o desde la aplicación Debian instalada sobre Windows (10) podemos clonar ese repositorio usando el comando:

git clone <https://github.com/mit-pdos/xv6-public>

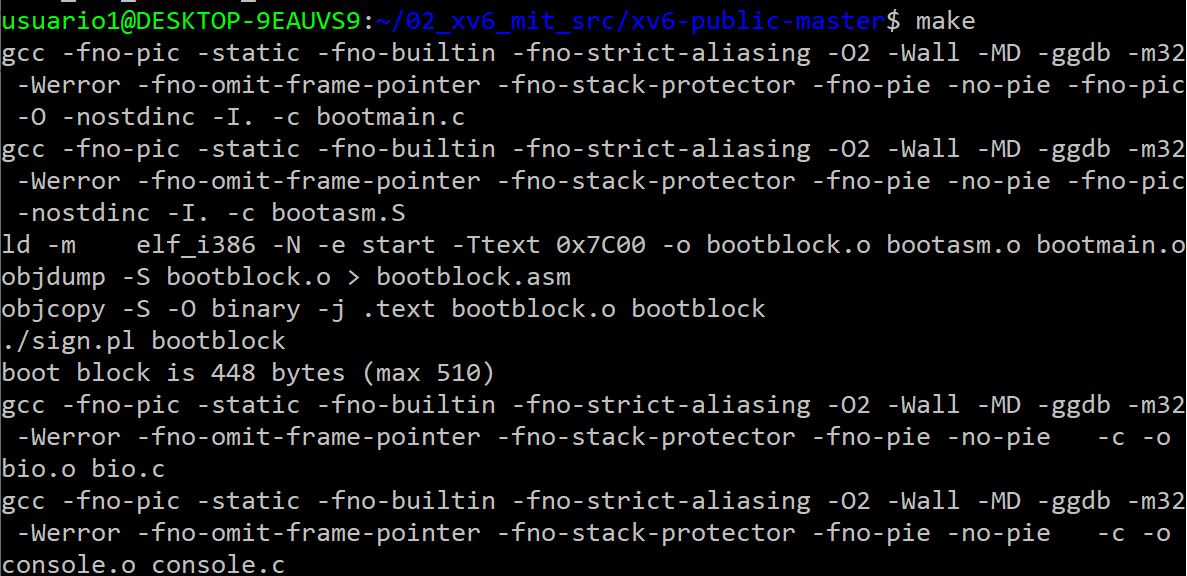
La salida debe ser algo como lo que se muestra:

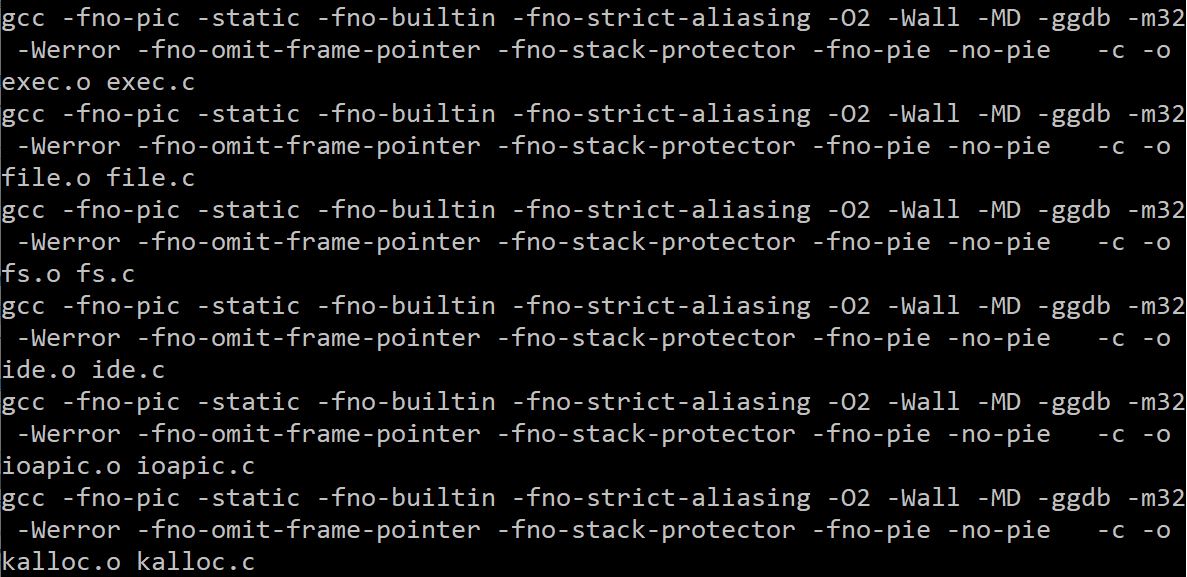


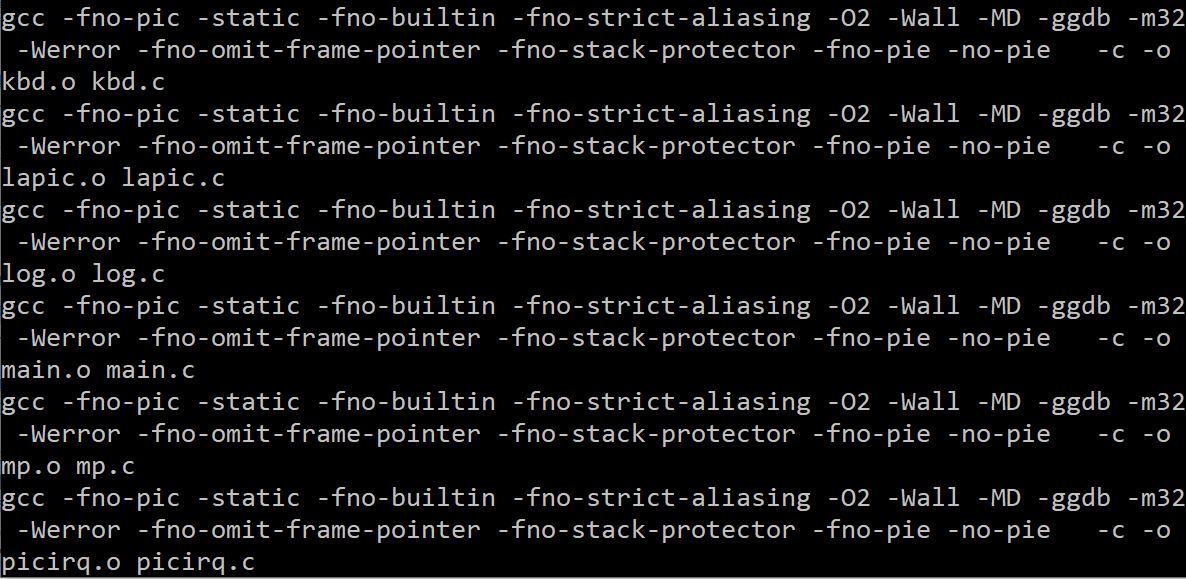
Clonación del repositorio xv6-public

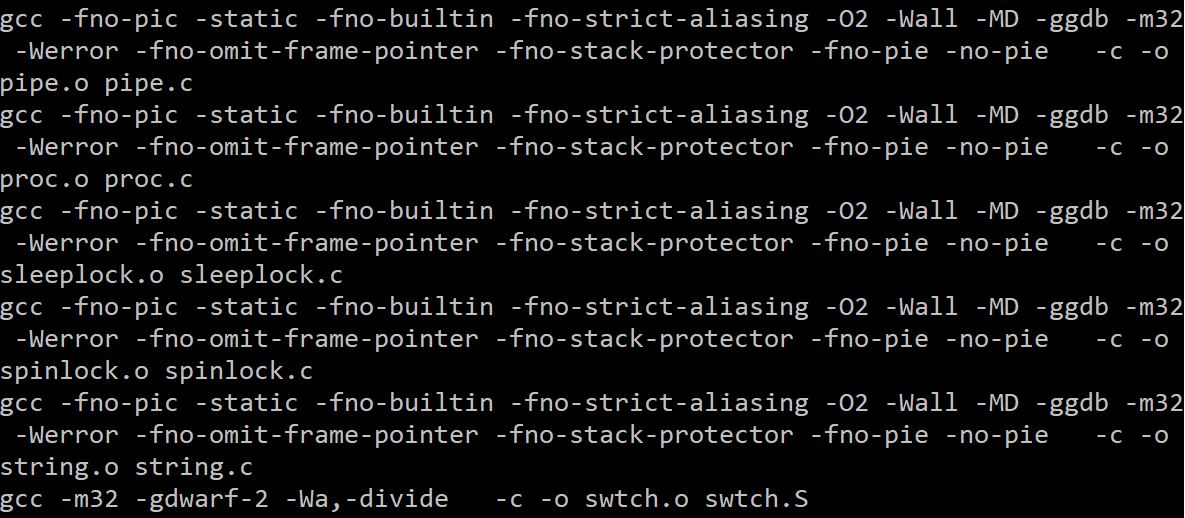
Después debemos ingresar al directorio xv6-public, el cual es una copia local del repositorio.

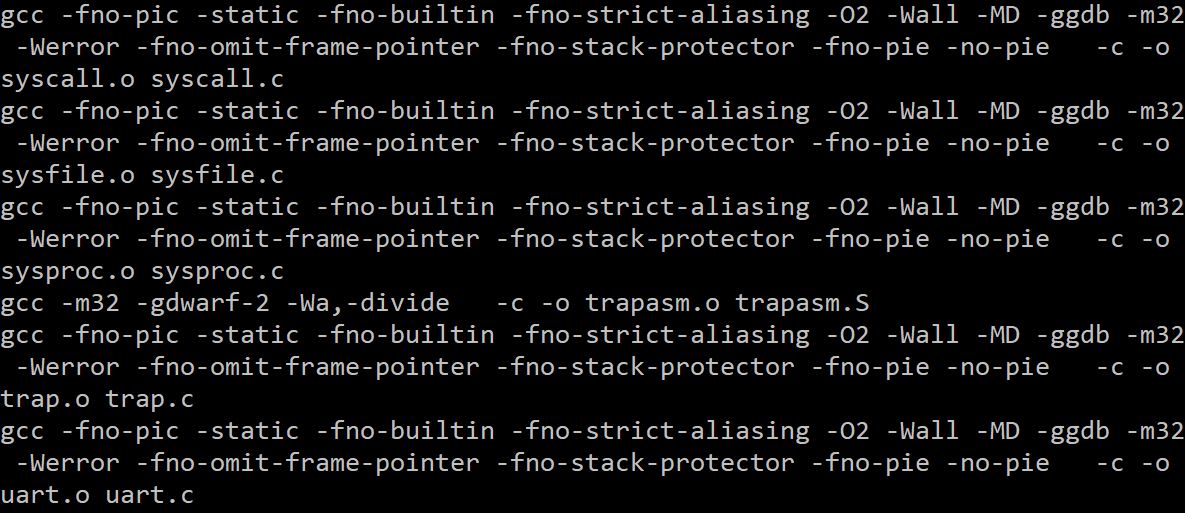
Dentro del directorio de fuentes de xv6, después de ejecutar el comando make se debe obtener las siguientes líneas de salida:

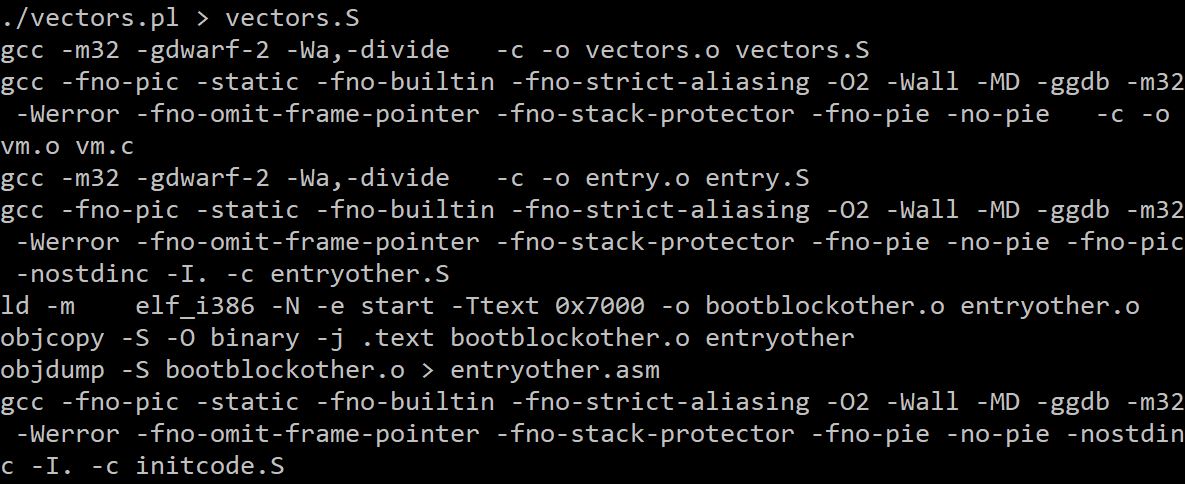


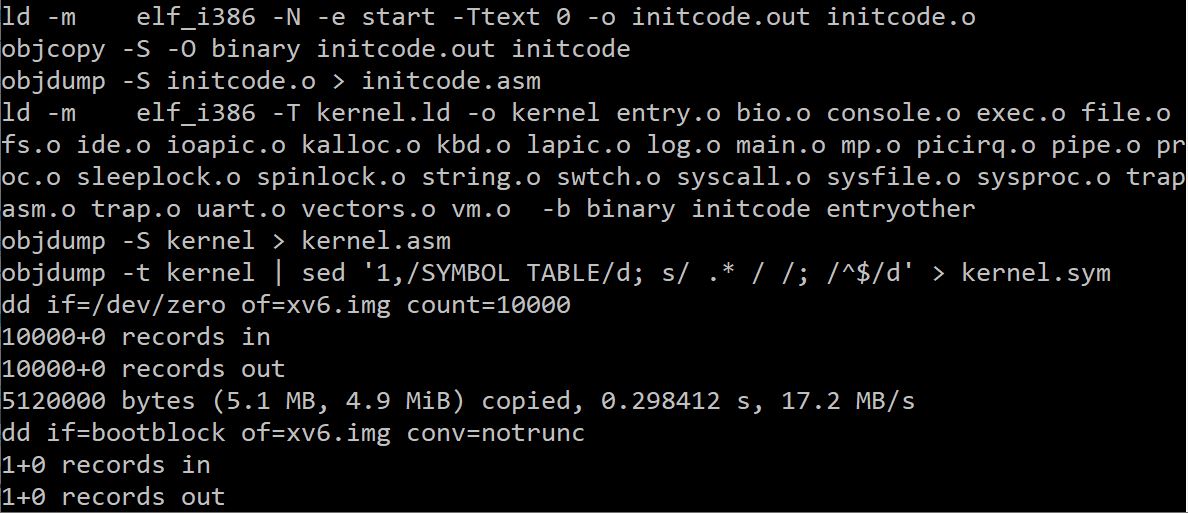


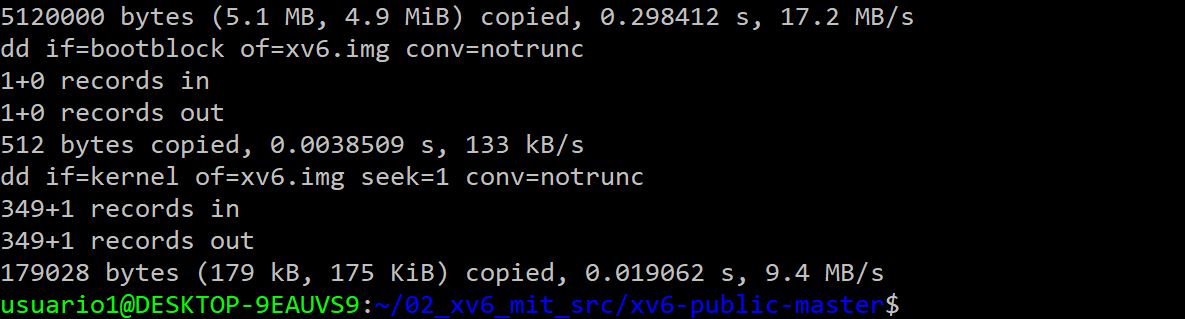








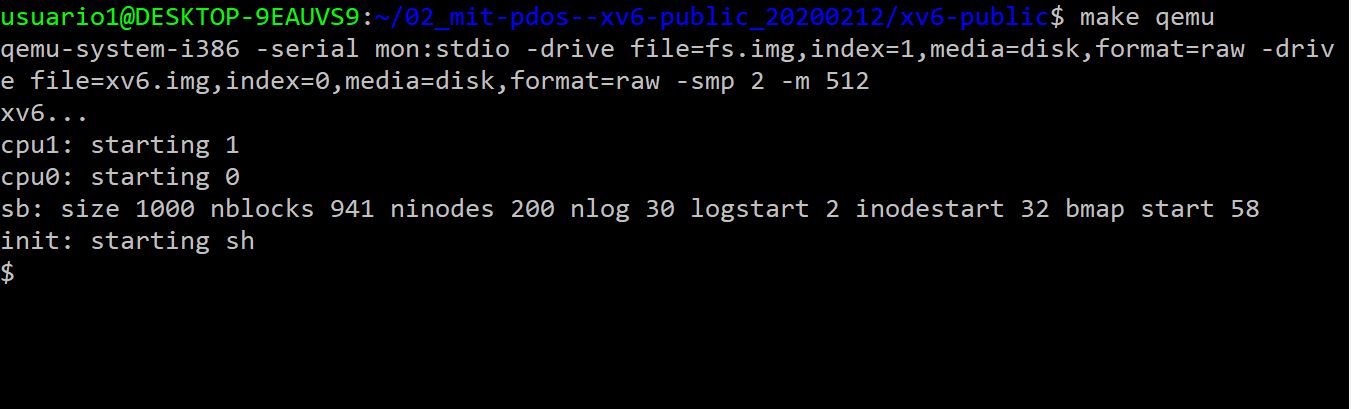




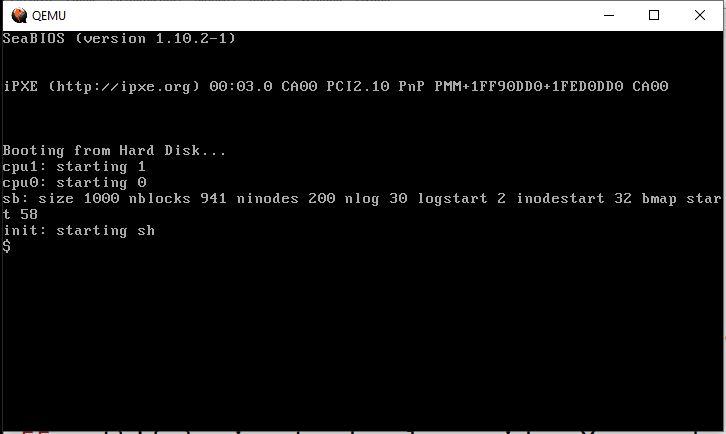
Ahora que ya está construido el sistema lo podemos ver funcionando solicitando la realización del objetivo qemu: para ello debemos teclear

make qemu

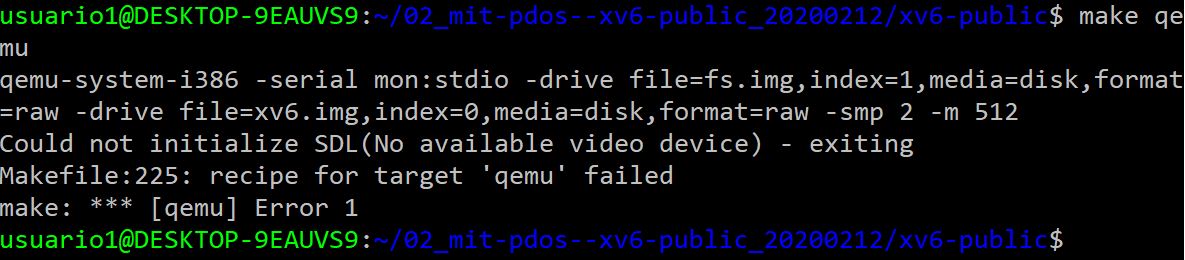
La salida correcta debe ser como se muestra en la siguiente imagen:



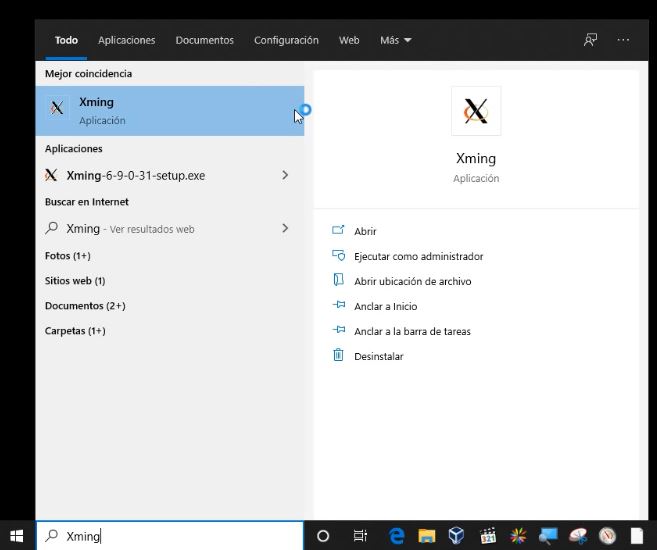
Y también se debe abrir la interfaz gráfica del qemu mostrando la ejecución del sistema xv6:



Si en lugar de lo anterior, se nos presenta el siguiente error:



Significa que no está corriendo el servidor X (el programa Xming), no hemos exportado el valor de la variable DISPLAY correctamente; o ambas cosas. Para corregir estos errores debemos poner a correr el servidor X, véase la siguiente imagen:



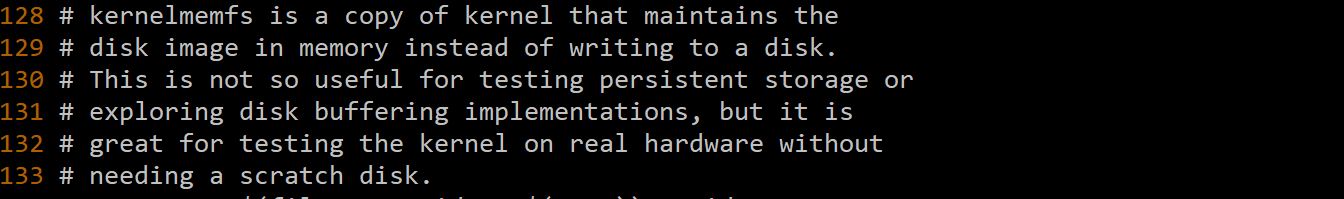
Una vez que estemos seguros de que ya se está ejecutando el servidor X, debemos exportar la variable de ambiente DISPLAY con el siguiente comando:

export DISPLAY=:0

Después de esto, el comando make qemu debería mostrar exitosamente la ejecución del sistema operativo xv6.

## Construcción de un kernel de xv6 para su ejecución sobre hardware real

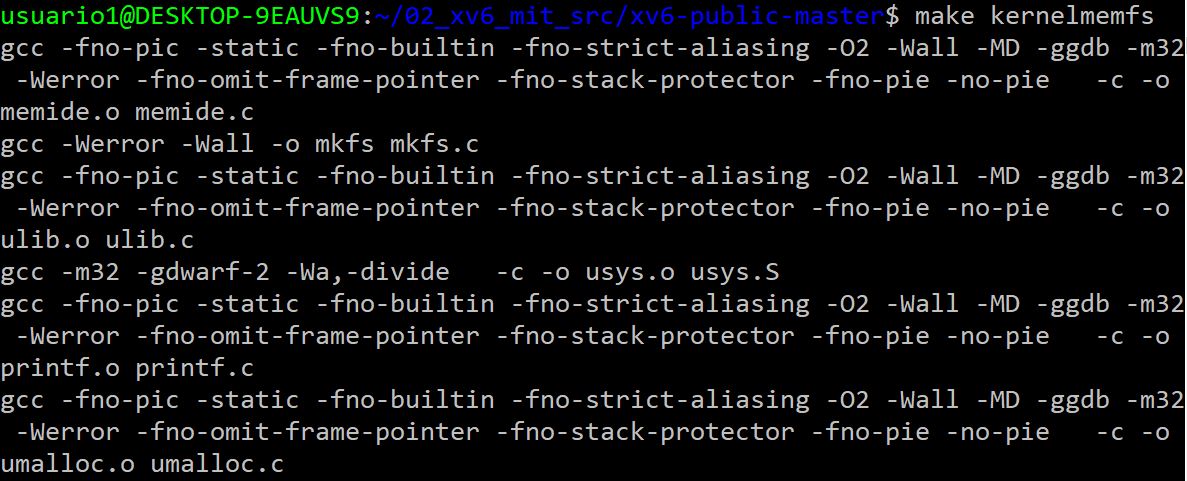
En el archivo Makefile de xv6 podemos leer lo siguiente:

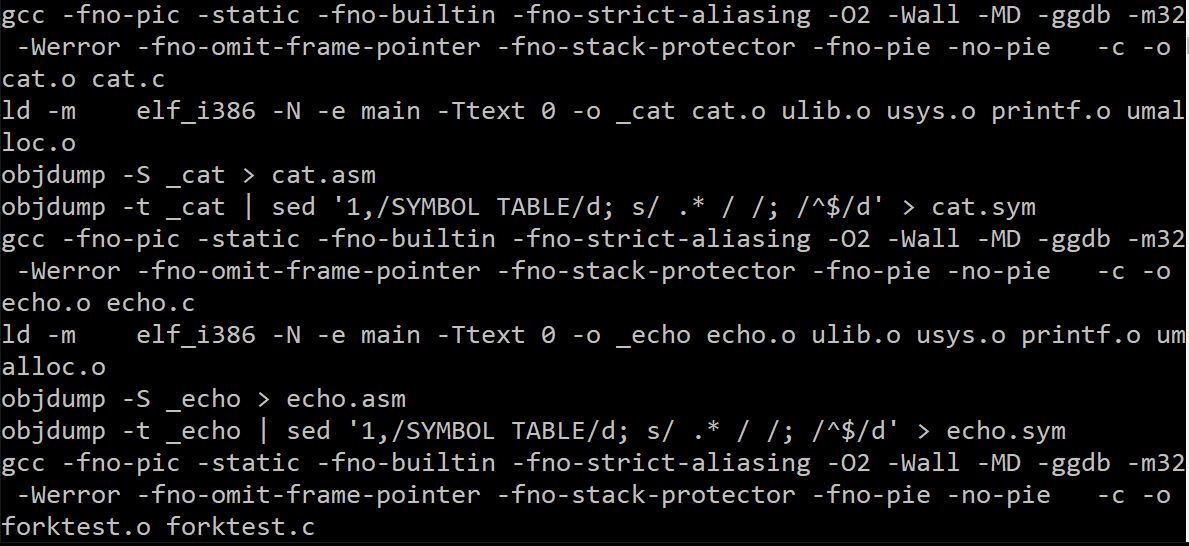


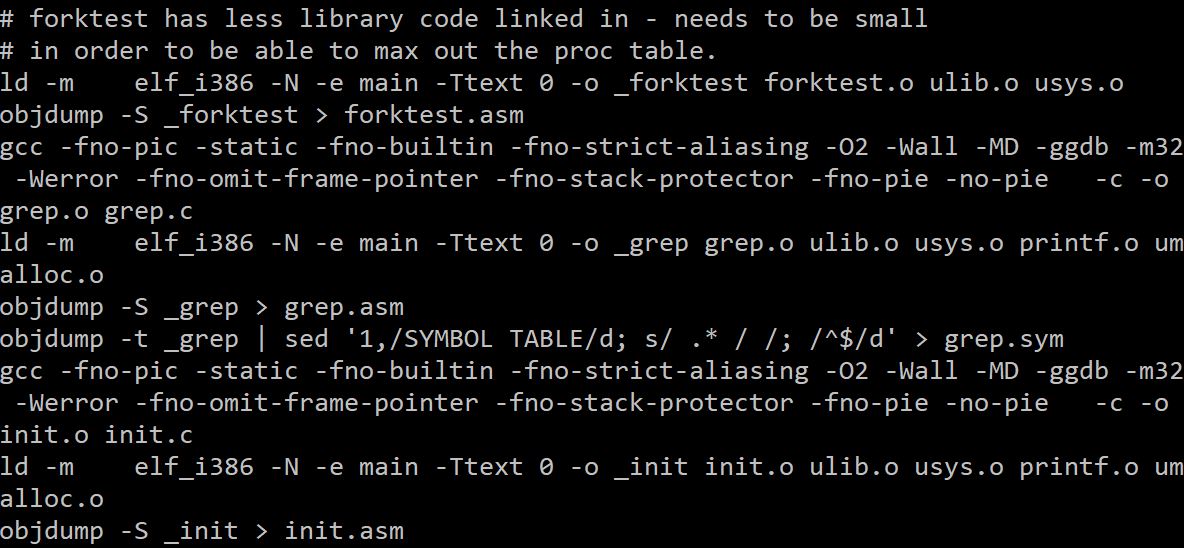
Por lo tanto, un kernel adecuado para intentar correr el xv6 sobre hardware real es kernelmemfs. Para construir el nucleo kernelmemfs, debemos ejecutar el comando:

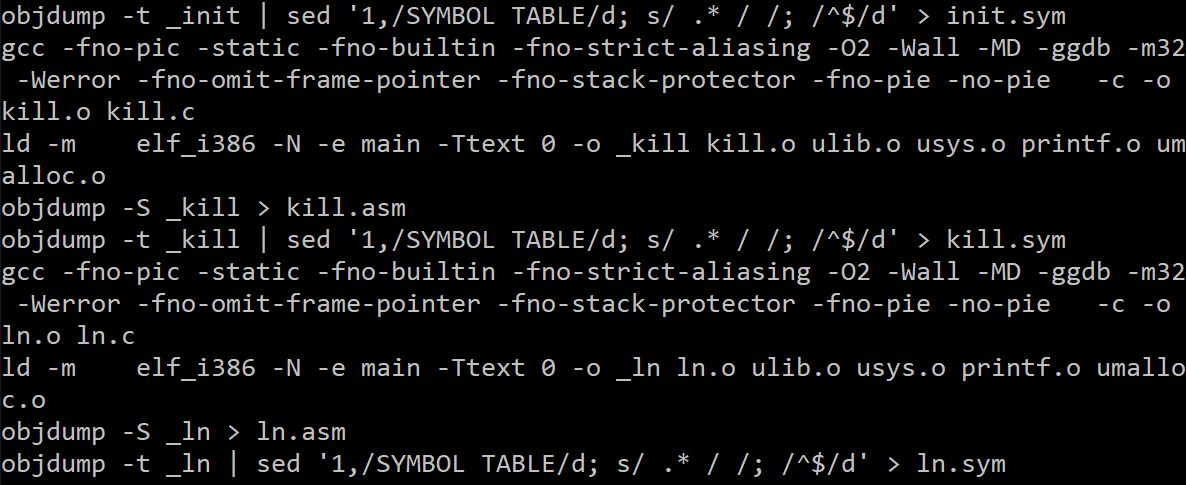
make kernelmemfs

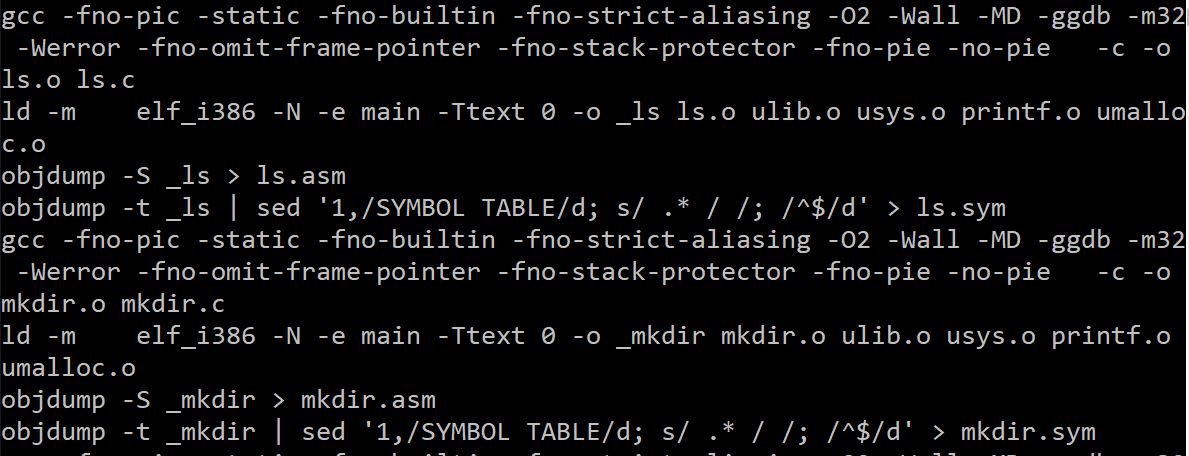
La salida correcta son muchas líneas como se muestra a continuación:

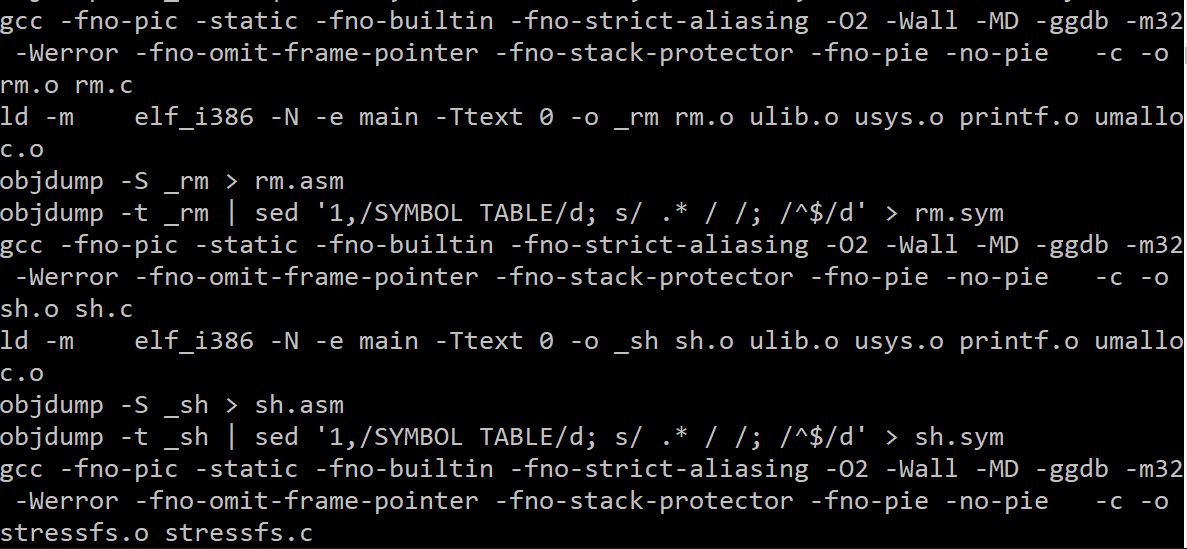


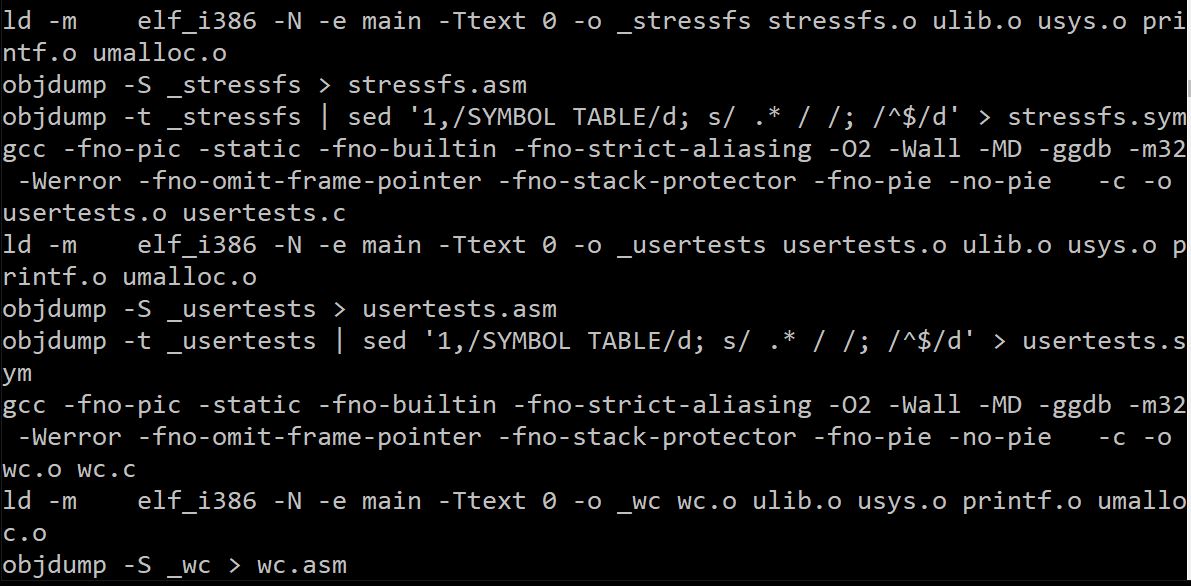


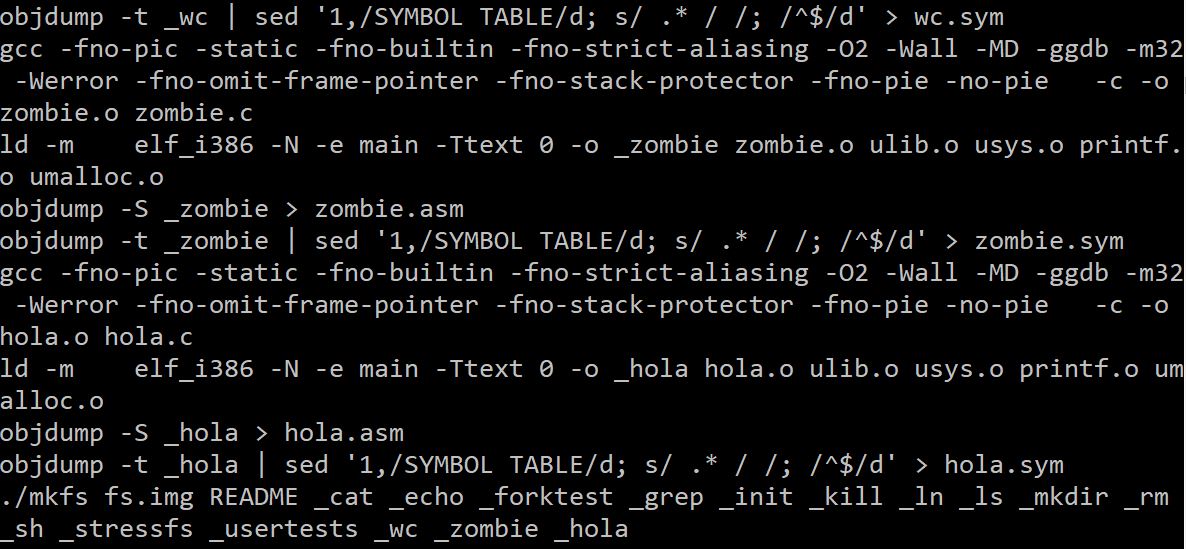


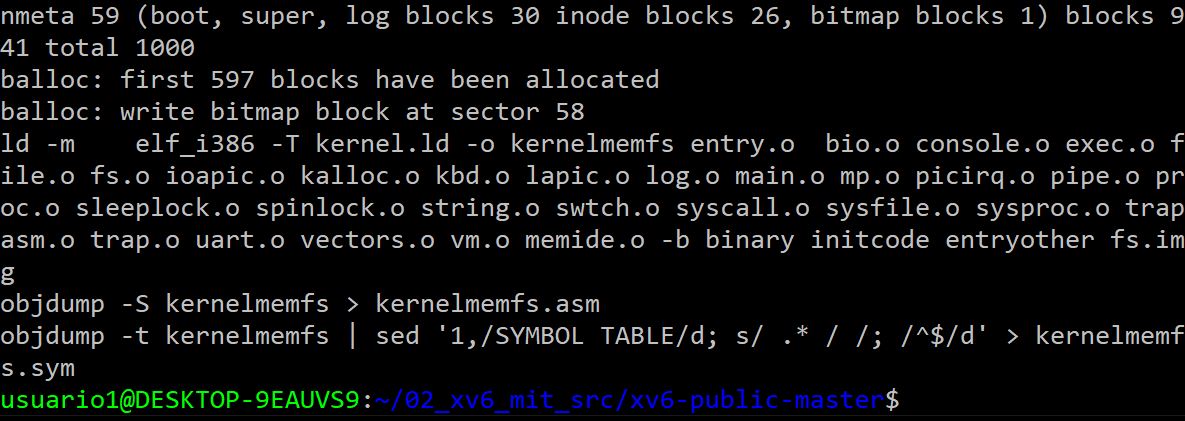








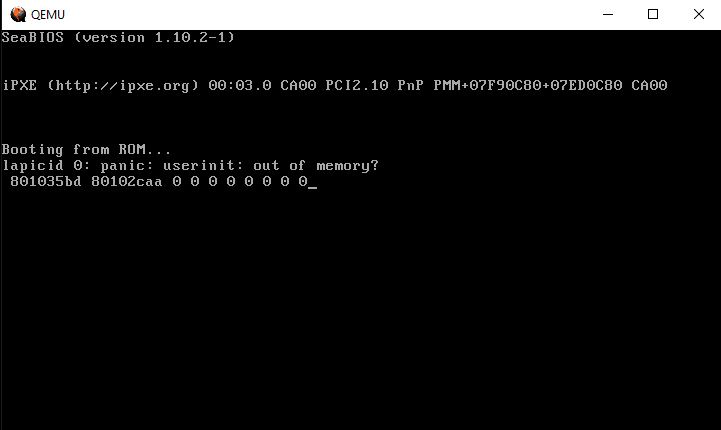




Ahora que ya se ha construido el nucleo kernelmemfs, podemos intentar ejecutaro sobre qemu con el siguiente comando:

qemu-system-i386 –kernel kernelmemfs

Es muy posible que se nos presente el siguiente error:

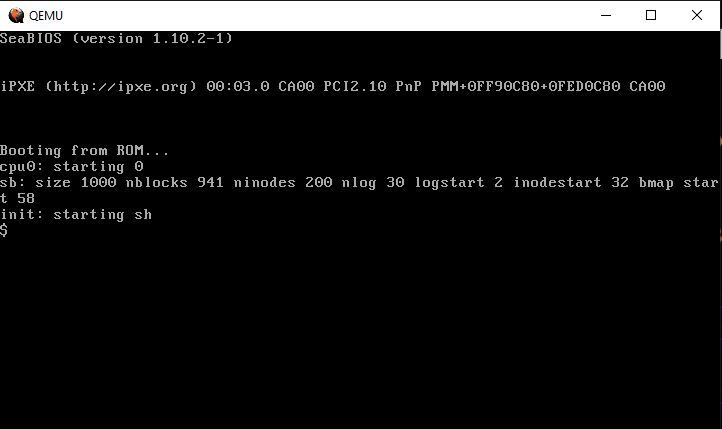


Error que se muestra en qemu cuando no se especifica –m 256

Entonces deberemos especificar una cantidad de memoria con el modificador –m del comando qemu:

qemu-system-i386 –kernel kernelmemfs –m 256

La salida correcta en el qemu debe ser algo como lo siguiente:



Salida correcta en qemu del comando: qemu-system-i386 –kernel kernelmemfs –m 256

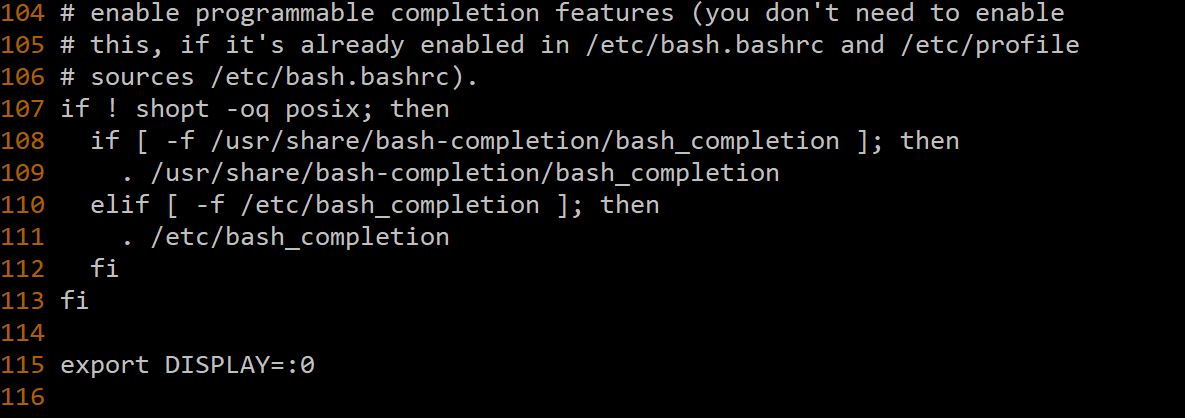
NOTA: si no queremos tener que estar exportando la variable DISPLAY

cada vez que lo necesitemos, podemos editar el archivo ~/.bashrc

para agregar al final de este el comando export DISPLAY=:0

Por ejemplo, por el momento las \'{u}ltimas l\'{i}neas de ese archivo

en mi instalaci\'{o}n de debian son las siguientes:



Despu\'{e}s de agregar la l\'{i}nea de "export DISPLAY=:0" al archiv ~/.bashrc podemos ejecutar el comando:

source ~/.bashrc

para que nuevamente se lea ese archivo de configuraci\'{o}n.

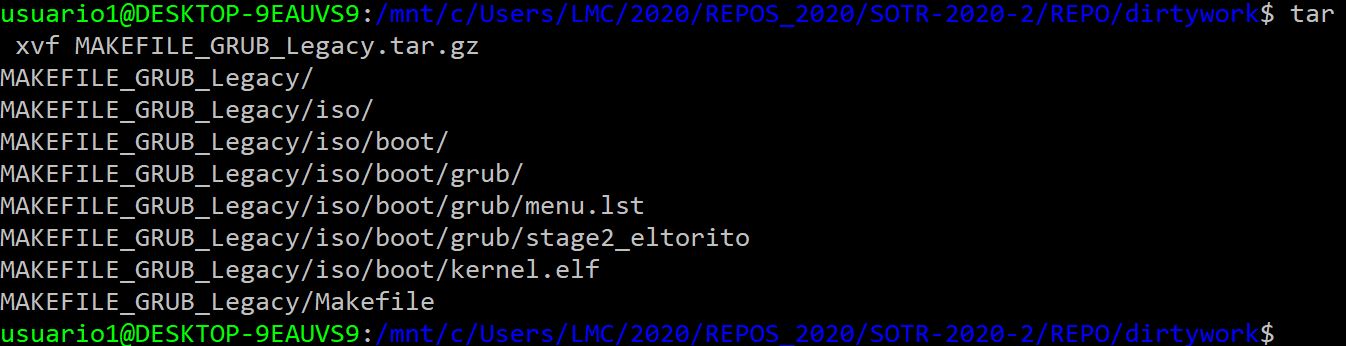
## Construcción de un archivo iso booteable con el sistema xv6

Para construir un archivo iso booteable a través del cual podremos ejecutar xv6 procederemos como sigue:

Descargar el archivo MAKEFILE\_GRUB\_Legacy.tar.gz del repositorio que estamos usando (<https://github.com/sotrteacher/dirtywork>). Una vez obtenido el archivo procedemos a extraer su contenido:

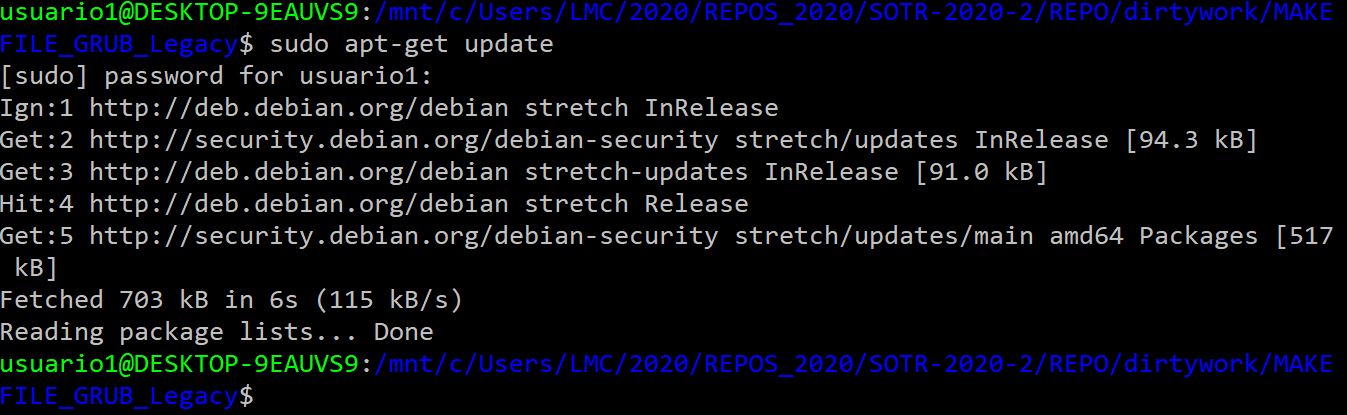
tar xvf MAKEFILE\_GRUB\_Legacy.tar

La salida debe ser:

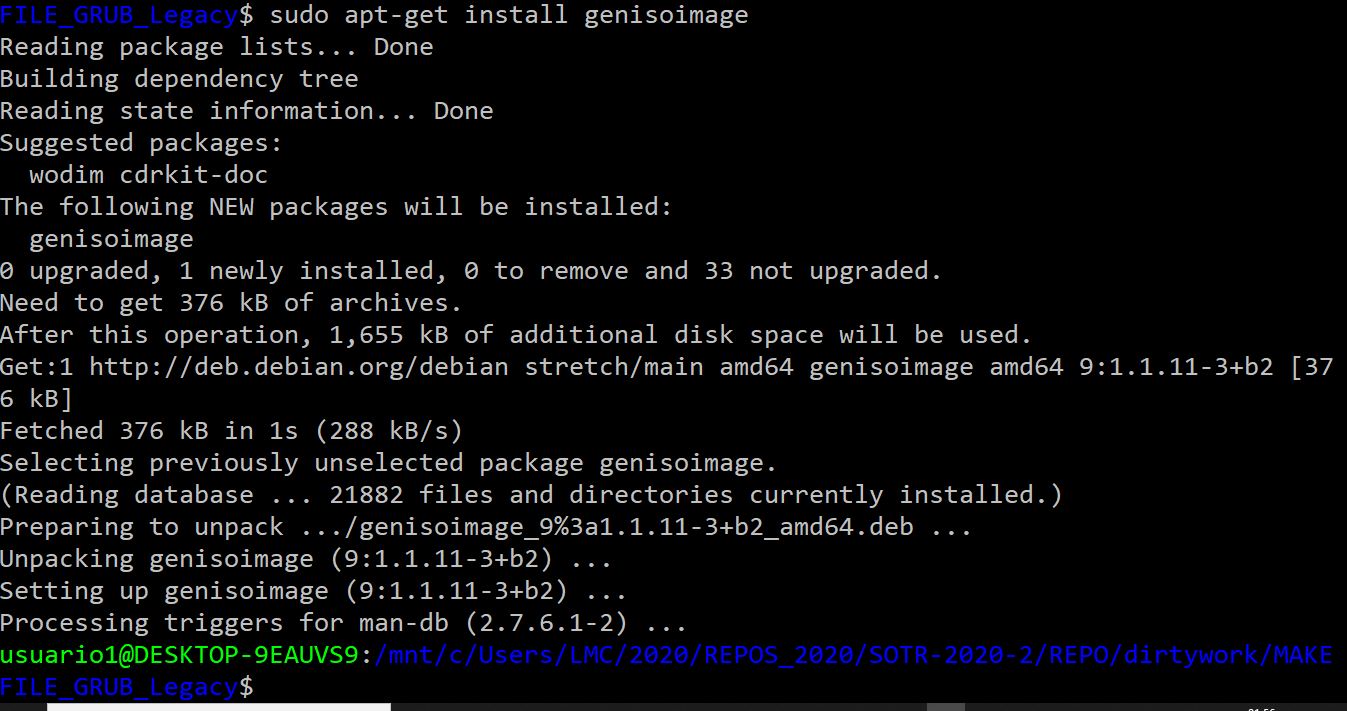


Antes de continuar, debemos asegurarnos de que tenemos instalado el programa genisoimage:

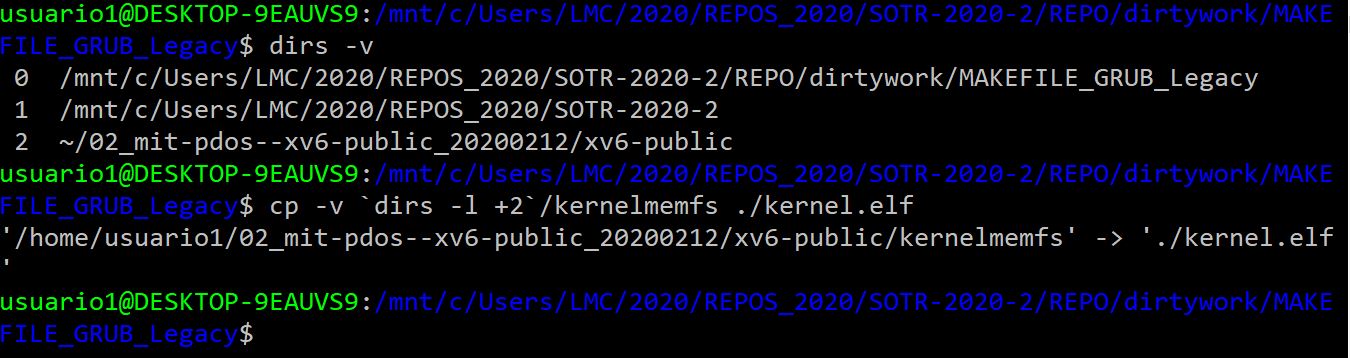
Actualizamos base de datos de paquetes



Instalamos el paquete genisoimage:

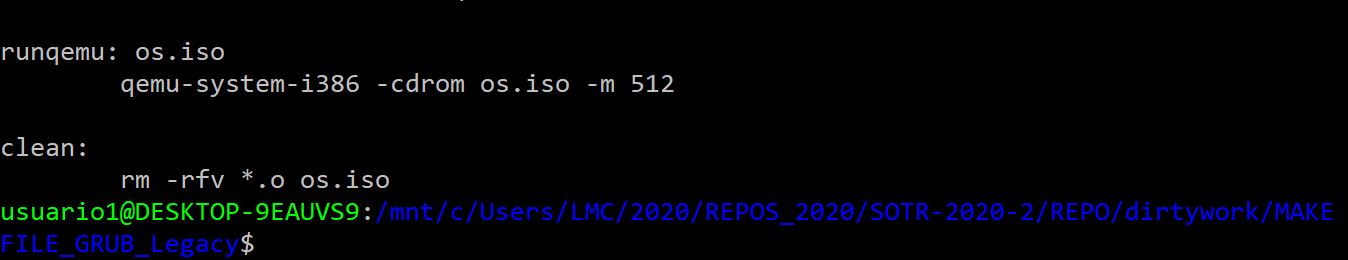


Ahora debemos poner una copia de nuestro nucleo kernelmemfs en el directorio MAKEFILE\_GRUB\_Legacy, pero con el nombre kernel.elf

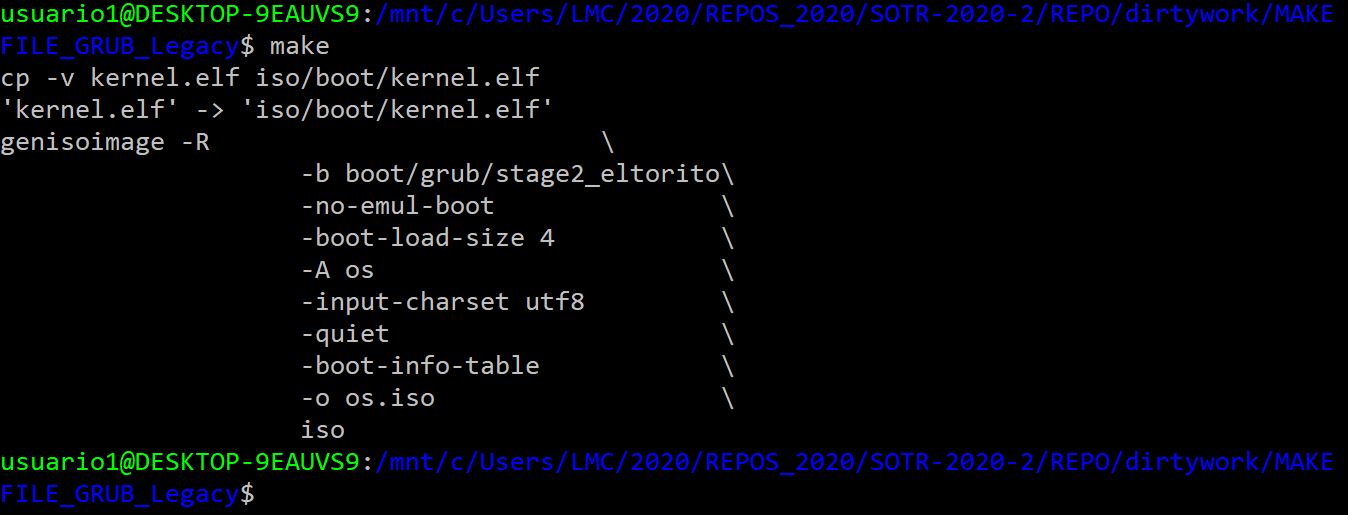


La razón de esto se puede encontrar en el contenido del archivo Makefile del directorio MAKEFILE\_GRUB\_Legacy, el cual se muestra aquí:





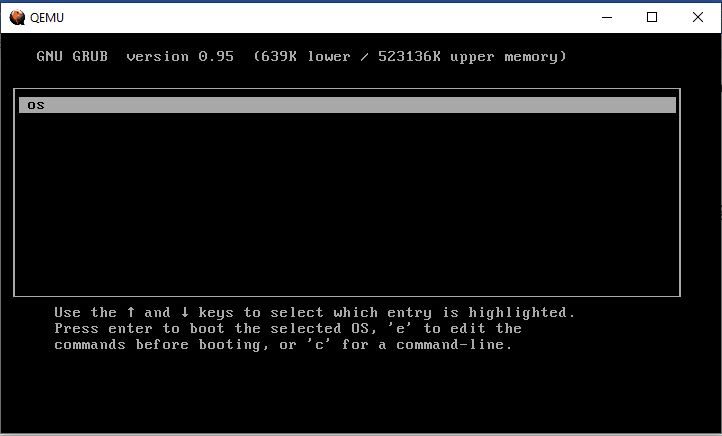
Una vez que hemos copiado el nucleo kernelmemfs con el nombre kernel.elf en el directorio MAKEFILE\_GRUB\_Legacy, para crear un archivo llamado os.iso solo teneos que teclear el comando make; la salida debe ser como se indica



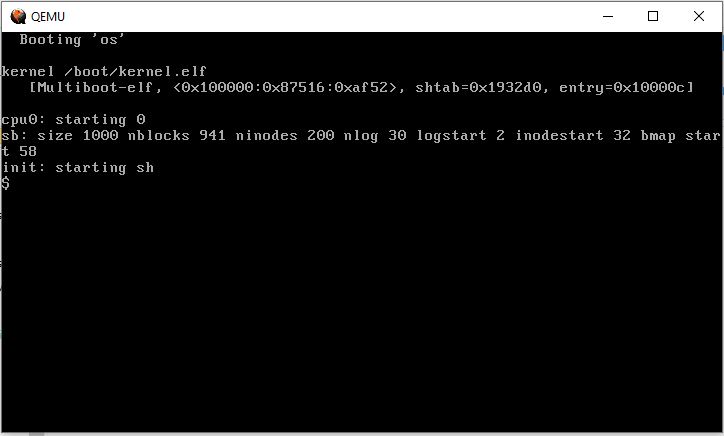
Ya con nuestro archivo os.iso recién creado podemos revisar su ejecución sobre quemu con el comando

make runqemu

Se debe observar durante 10 segundos en la interfaz gráfica de qemu algo como lo siguiente:



Una vez transcurridos los 10 segundos deberemos ver la ejecución del sistema xv6 como se muestra en la siguiente imagen:



Con esto, comprobamos que nuestra imagen iso fue construida correctamente.

## Creación de un Live-usb para correr xv6 con el nucleo kernelmemfs

El siguiente paso es crear un live-usb para probar la ejecución de xv6 sobre un hardware real. Primero ejecutamos el programa Linux Live USB Creator:

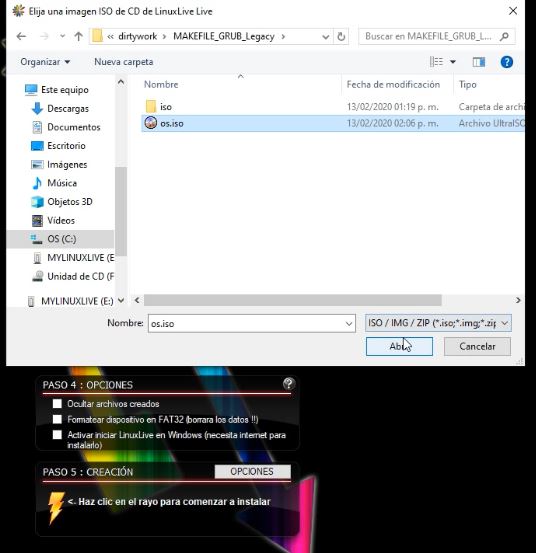


Debemos seleccionar la USB en el combo box que dice ->Elija un dispositivo:



Como segundo paso, debemos seleccionar nuestro archivo ISO:





Por último, damos clic en el botón que tiene la imagen de un rayito:



Si todo va bien, deberemos ver (luego de unos minutos) el siguiente mensaje:



Después de esto tendremos un live-usb que podemos usar para intentar correr el sistema operativo xv6 con su nucleo kernelmemfs sobre un hardware real.