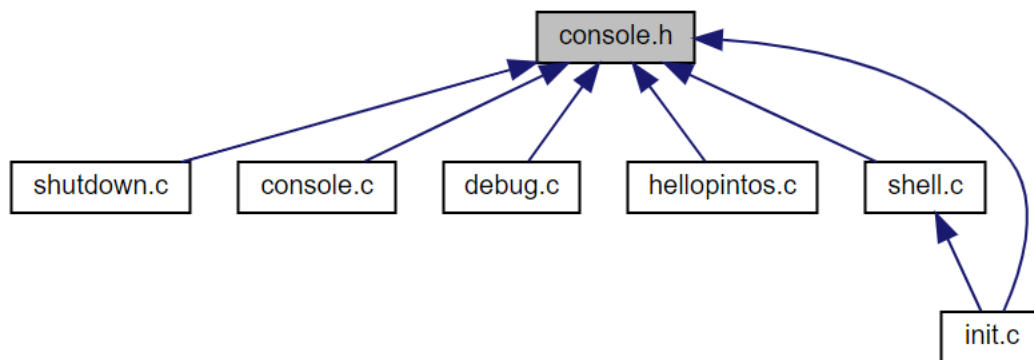


A) Tabla con las componentes relevantes para el proyecto.

Componentes	Descripción
/pintos/src/tests/threads/hellopintos.c	En este fichero se define el <i>test</i> de un hola mundo para ser ejecutado en pintos. Una vez ejecutado este test se produce el <i>output</i> “Hello, pintos” sin las comillas.
/pintos/src/tests /threads/tests.c	En este fichero se encuentran una serie de <i>tests</i> que serán ejecutados dependiendo de los parámetros que sean enviados cuando se ejecute la sentencia “pintos – run [test]”
/pintos/src/tests/threads/tests.h	En este fichero se encuentran los prototipos de las funciones que son consideradas <i>tests</i> .
/pintos/src/threads/init.c	Aquí inicia el kernel de Pintos, lo que permite que se reciban los parámetros que darán lugar al programa que será ejecutado una vez el Sistema Operativo suba.
/pintos/src/threads/thread.c	Este archivo se encarga de crear el <i>thread</i> que aloja el programa que se ejecutará (especificado en el comando que da inicio a Pintos).
/pintos/src/tests/threads/make.tests	Archivo en el que se definen los pasos para compilar el archivo hellopintos.c

B) Grafo de todas las dependencias de las componentes relevantes para el desarrollo de la aplicación.**C) Locus para el desarrollo de la aplicación solicitada. Debe aparecer la fundamentación de la elección.**

El locus que elegí fue *pintos/src/tests/threads/hellopintos.c*. Esta elección se debe a que por defecto es donde se encuentran los tests que vienen predefinidos y no vi ninguna razón conveniente que justificara el uso de uno diferente. También es de notar que en este directorio se encuentra un archivo *Makefile* que define los pasos para la compilación de los tests, lo cual es otra razón a favor del uso de este directorio.

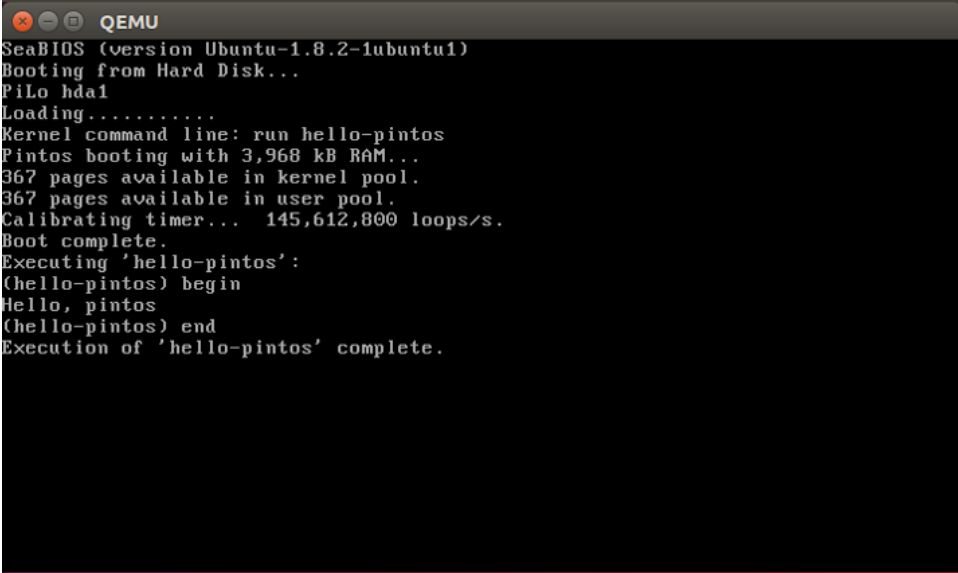
D) Indicaciones de producción del ejecutable y la documentación

Para la producción adecuada del ejecutable es necesario ir al directorio */pintos/src/threads*, allí se abre una instancia de la terminal y se procede a hacer un *make clean* seguido de *make all*, esto para asegurar que se está trabajando con la versión más reciente de los archivos del proyecto.

Una vez estos comandos fueron ejecutados se puede proceder a ejecutar el comando – **pintos – run “hello-pintos”** el cual, en caso de tener todo lo necesario instalado, abrirá una instancia del emulador por defecto, en este caso QEMU, en el cual cargará PintOS para posteriormente imprimir el mensaje “Hello, pintos” e inmediatamente concluir con la ejecución de PintOS.

Por otro lado, la documentación se produce desde el directorio */pintos* ejecutando el comando “doxygen Doxygen” donde *Doxygen* es el nombre del archivo donde se incluyen los elementos de configuración para la producción del archivo html que se genera.

E) Captura de pantalla del resultado del ejecutable. Debe ir acompañada de una explicación de su contenido.



```
SeaBIOS (version Ubuntu-1.8.2-1ubuntu1)
Booting from Hard Disk...
PiLo hda1
Loading.....
Kernel command line: run hello-pintos
Pintos booting with 3,968 kB RAM...
367 pages available in kernel pool.
367 pages available in user pool.
Calibrating timer... 145,612,800 loops/s.
Boot complete.
Executing 'hello-pintos':
(hello-pintos) begin
Hello, pintos
(hello-pintos) end
Execution of 'hello-pintos' complete.
```

En esta captura de pantalla se puede ver el resultado de la ejecución del comando **pintos – run hello-pintos** el cual ejecuta el test contenido en `helloworld.c` que consiste en imprimir el mensaje “Hello, pintos”, una vez esto finaliza la ejecución de Pintos finaliza.