

# Configuración del entrenamiento

*Crear la estructura de directorios utilizados en los laboratorios prácticos*

Todos los laboratorios se realizaron utilizando la versión 18.04 de Ubuntu. Es posible que en una versión anterior o posterior se necesiten instalar distintos paquetes o cambien algunas de las configuraciones. Lo mismo aplica a distribuciones diferentes de GNU/Linux.

## Configuración del espacio de trabajo del proyecto

Cree el espacio de trabajo, a través de los siguientes comandos basados en 'Building embedded Linux Systems' (K. Yaghmour):

```
export PROJECT=dev_bbb_glibc_labs
export PROJECT_ROOT=`pwd`/${PROJECT}

mkdir -p ${PROJECT_ROOT}/{bootloader/data,build-tools/data,download,\
doc,images,kernel/data,project,rootfs,sysapps/data,tmp,tools,bin}

cat > ./dev_bbb_glibc << "EOF"
export PROJECT=dev_bbb_glibc_labs
export PROJECT_ROOT=`pwd`/${PROJECT}
export ARCH=arm
export TARGET=arm-cortex_a8-linux-gnueabi
export CROSS_COMPILE=${TARGET}-
export PREFIX=${PROJECT_ROOT}/tools
export TARGET_PREFIX=${PREFIX}/${TARGET}
export PATH=${TARGET_PREFIX}/bin:${PROJECT_ROOT}/bin:${PATH}
cd ${PROJECT_ROOT}
EOF

. dev_bbb_glibc
```

Asegurese que los parámetros se asignaron de forma correcta!!! (echo \$PROJECT\_ROOT)

De ahora en adelante, antes de comenzar a trabajar en los laboratorios, puede iniciar las variables de entorno necesarias ejecutando el shell script dev\_bbb\_glibc.

## Más pautas

Pueden ser útil en cualquiera de los laboratorios.

- Lea las instrucciones y consejos cuidadosamente. Muchas personas cometen errores o pierden tiempo porque se han perdido una explicación o una guía.
- Lea siempre cuidadosamente los mensajes de error, en particular el primero que se emite. Algunas personas tropiezan con errores muy simples simplemente porque especificaron

una ruta de archivo incorrecta y no prestaron suficiente atención al mensaje de error correspondiente.

- Nunca te quedes atascado con un problema extraño más de 5 minutos. Muestra tu problema a tus colegas o al instructor.
- Solo debe usar el usuario `root` para las operaciones que requieren privilegios de superusuario, tales como: montar un sistema de archivos, cargar un módulo del kernel, cambiar la propiedad del archivo, configurar la red. La mayoría de las tareas regulares (como descargar, extraer fuentes, compilar ...) se pueden realizar como usuarios regulares.
- Si ejecutó comandos desde un shell raíz por error, es posible que su usuario habitual ya no pueda manejar los archivos generados correspondientes. En este caso, use el comando `chown -R` para devolver los nuevos archivos a su usuario habitual.  
Ejemplo: `chown -R myuser.myuser linux-3.4`