**Tecnológico de Costa Rica**

**Arquitectura de Computadoras**

**Profesor: Dr. Johan Carvajal**

**Desarrollado por:**

* Tania Martínez Abarca
* Roy Moya Serrano
* David Picado Aguilar
* German Quesada Solís

**Descripción de documento**

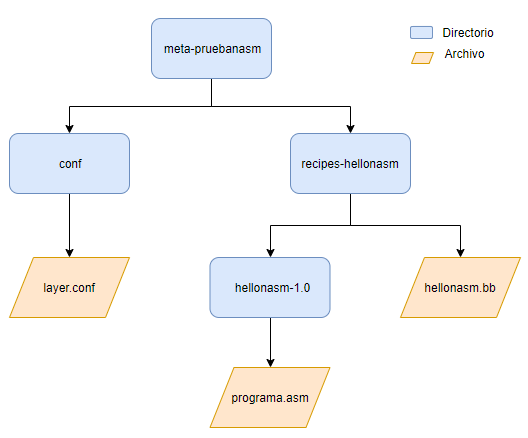
El presente documento es una guía para el lector con el fin de seguir una serie de pasos para el reconocimiento del lenguaje ensamblador Nasm en Yocto. Propiamente, se indicarán las carpetas a las cuales se les realizaron cambios, y se ejemplificará con un programa de Nasm (.asm) para generar un “Hola Mundo”; además de esto mostrar la manera en la que se debió compilar para poder ser reconocido por Yocto. Los cambios hechos en estas carpetas se pueden encontrar en el repositorio de github denominado “TRDG”.

**Pasos a seguir**

1. Ubicados en la carpeta de poky-warrior se procede a ingresar a la carpeta denominada “meta” y una vez dentro de esta se ingresa a la carpeta “conf”. En la carpeta de “conf” se encontrará el archivo “bitbake.conf”, en este se agregará el ensamblador nasm colocando su nombre en la sección llamada HOSTTOOLS.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | # Tools needed to run builds with OE-Core | |  | # python is special cased to point at python2 | |  | HOSTTOOLS += " \ | |  | [ ar as awk basename bash bzip2 cat chgrp chmod chown chrpath cmp comm cp cpio \ | |  | cpp cut date dd diff diffstat dirname du echo egrep env expand expr false \ | |  | fgrep file find flock g++ gawk gcc getconf getopt git grep gunzip gzip \ | |  | head hostname id install ld ldd ln ls make makeinfo md5sum mkdir mknod \ | |  | mktemp mv nm nasm objcopy objdump od patch perl pod2man pr printf pwd python2 \ | |  | python2.7 python3 ranlib readelf readlink realpath rm rmdir rpcgen sed seq sh sha256sum\ | |  | sleep sort split stat strings strip tail tar tee test touch tr true uname \ | |  | uniq wc wget which xargs \ | |  | " | |

1. Se inicializa la carpeta pocky-warrior con el comando “source oe-init-build-env”, para que se reconozca el cambio en la carpeta mencionada en el punto 1.
2. Seguidamente se usa el comando “bitbake nasm” para que compile el cambio hecho en el HOSTTOOLS.
3. Ahora se va a crear la nueva layer o capa que va a contener el archivo de nasm. Se utilizará una carpeta de prueba a la cual se le llamará “meta-pruebanasm”, en ella se agregarán la receta y la configuración. En el siguiente diagrama se mostrará la manera en que se divide la carpeta de prueba.



1. El archivo “layer.conf” tendrá la configuración de la layer que estamos creando y su contenido será el siguiente:

|  |
| --- |
| # We have a conf and classes directory, add to BBPATH |
| BBPATH .= ":${LAYERDIR}" |
|  |
| # We have recipes-\* directories, add to BBFILES |
| BBFILES += "${LAYERDIR}/recipes-\*/\*.bb \ |
| ${LAYERDIR}/recipes-\*/\*/\*.bbappend" |
|  |
| BBFILE\_COLLECTIONS += "pruebanasm" |
| BBFILE\_PATTERN\_pruebanasm = "^${LAYERDIR}/" |
| BBFILE\_PRIORITY\_pruebanasm = "5" |
| LAYERVERSION\_pruebanasm = "4" |
| LAYERSERIES\_COMPAT\_pruebanasm = "warrior" |

1. La carpeta de recipes contiene el archivo que se quiere reconocer en Yocto, en este caso el “programa.asm”, que está situado dentro de la carpeta de la versión de la recipe “hellonasm-1.0”. En este código se utilizan registros de 32 bits donde “int 80h” representa “syscall” (Se compila con gcc).

|  |
| --- |
| section .data  hello: db "Hello World!",10 |
| helloLen: equ $-hello |
| section .text |
| global main |
| main: |
| mov eax, 4 |
| mov ebx, 1 |
| mov ecx, hello |
| mov edx, helloLen |
| int 80h |
|  |
| mov eax, 1 |
| mov ebx, 0 |
| int 80h |

Contenido de programa.asm

1. Esta carpeta de recipes también se conforma por el archivo “hellonasm.bb”, el cual contiene la descripción de la receta (manera en la que se va a reconocer el archivo .asm, cómo se compila y cómo se linkea). El contenido del archivo es:

|  |
| --- |
| LICENSE = "CLOSED" |
| LIC\_FILES\_CHKSUM = "" |
| # No information for SRC\_URI yet (only an external source tree was specified) |
| SRC\_URI = "file://programa.asm" |
| # NOTE: no Makefile found, unable to determine what needs to be done |
| S = "${WORKDIR}" |
| TARGET\_CC\_ARCH += "${LDFLAGS}" |
| do\_configure () { |
| # Specify any needed configure commands here |
| : |
| } |
| do\_compile () { |
| nasm -f elf programa.asm -o programa.o |
| ${CC} -o programa programa.o |
| } |
| do\_install () { |
| install -d ${D}${bindir} |
| install -m 0755 programa ${D}${bindir} |
| } |

1. Una vez creada la layer deseada, se procede a modificar ciertos archivos dentro de la carpeta “poky-warrior”. El primer archivo a modificar es “bblayers.conf” para agregar esta nueva layer, el archivo se encuenta en build/conf.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | POKY\_BBLAYERS\_CONF\_VERSION is increased each time build/conf/bblayers.conf  # changes incompatibly | | POKY\_BBLAYERS\_CONF\_VERSION = "2" | |  | | BBPATH = "${TOPDIR}" | | BBFILES ?= "" | |  | | BBLAYERS ?= " \ | | /home/tania/Escritorio/poky-warrior/meta \ | | /home/tania/Escritorio/poky-warrior/meta-poky \ | | /home/tania/Escritorio/poky-warrior/meta-yocto-bsp \ | | /home/tania/Escritorio/poky-warrior/meta-pruebanasm \ | |

1. EL otro archivo a modificar es “local.conf” para agregar la nueva receta generada.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | IMAGE\_INSTALL\_append = " \ | |  | hellonasm \ | |  | " | |

1. Una vez que se hicieron todos estos cambios, se inicializa la carpeta de poky-warrior con el comando “source oe-init-build-env”, y una vez inicializado se genera el bitbake de la receta “bitbake hellonasm”.
2. Seguido se puede volver a inicializar el sistema si se desea con el mismo comando mencionado, y una vez inicializado se realiza el “bitbake core-image-minimal” (se utiliza minimal porque esa fue la que se utilizó al momento de descargar la carpeta de poky-warrior).
3. Por último se coloca el comando “runqemu qemux86” para cargar la imagen customizada en Linux.