Yocto Project

Topic: Yocto

Time: 2 Hours **Marks:**50

Part A & Part B all questions are mandatory.

Part-A 20*1=20

1.Write a command to see the workdir of core-image-minimal.

Answer: bitbake -e core-image-minimal | grep "^WORKDIR="

2.Who defines the do_fetch, do_config and other tasks.

Answer:The Yocto Project's BitBake build system defines the "do_fetch," "do_configure," and other tasks.

3.Where do I find build logs.

Answer:Build logs can typically be found in the "log" directory within your Yocto Project build directory.

4.What is the default behaviour of do_compile task.

Answer:The default behavior of the "do_compile" task in the Yocto Project is to compile the source code of the recipe.

5. What is the default behaviour of do_configer task.

Answer:The default behavior of "do_configure" is to run the auto-configuration process for the source code.

6.What is the purpose of SRCREV variable in a recipe.

Answer:The purpose of the SRCREV variable in a Yocto Project recipe is to specify a specific revision (commit) of the source code repository that should be used for building the recipe.

7.If we are fetching a source file from local git repository what change we made in SRC_URI variable.

Answer:If you are fetching a source file from a local Git repository, you should set the SRC_URI variable to specify the local Git repository path instead of a remote URL.

8.Write command to generate tag in local git repository.

Answer: git tag <tagname>

9.Write a git command to add any remote branch in your local repository.

Answer: git checkout -t origin/remote-branch-name

10.In a recipe what changes in WORKDIR when we include readme.txt file in includedir.

Answer: do_install() {

```
install -m 0644 ReadMe.txt ${D}${docdir}
```

11.Write a command to create static library from object file usr.o

Answer: ar rcs libexample.a usr.o

12.How to create a alies name of any recipe.

Answer:To create an alias name for a recipe in the Yocto Project, you can use the PN variable in the recipe's .bb file. Set the PN variable to the desired alias name. PN = "my-alias-name"

13.How to compile below program for getting output: hello system call.

```
#include<stdio.h>

int main()

{

#ifdef USE_SYSCALL

write(1,”hello system call”,17);

#else

printf(“hello world\n”);

#endif

return 0;

}
```

Answer:gcc test.c -o test -DUSE_SYSCALL

14.What is the purpose of ALLOW_EMPTY:${PN}=’1’ in a recipe.

Answer:The ALLOW_EMPTY:${PN}='1' in a Yocto Project recipe is used to indicate that the recipe can be empty without causing a build error.

15.If we write inherit autotools in a recipe which file it will inherit and what is the location of that file.

Answer:When you write inherit autotools in a Yocto Project recipe, it instructs the build system to inherit the "autotools" class.

Location: poky/meta/classes/autotools.bbclass

16.What is the use of local and autoconf command in autotools.

Answer: autoconf: The autoconf command is a tool used to automatically generate the configure script from the configure.ac or configure.in script.

local: The local command is used to define and set local shell variables within the configure.ac or configure.in script.

17.What is devshell? How to get a devshell.

Answer:A "devshell" in the Yocto Project is a development environment that allows you to work within an isolated and controlled environment, typically inside a target's root file system.

Command: bitbake -c devshell <recipe>

18.In a recipe how to add a folder other than file in a FILESPATH variable.

Answer: FILESPATH_prepend = "${THISDIR}/myfolder:"

19.How to validate weather my bbappend is successfully applied to recipe.

Answer: bitbake -e <recipe> | grep "^INHERIT"

20.If we write inherit kernel in a recipe which file it will inherit and what is the location of the file.

Answer: When you write inherit kernel in a Yocto Project recipe, it inherits the kernel-related classes and functionality from the Poky reference system.

Location: poky/meta/classes/kernel.bbclass

Part-B 15*2=30

1.What is layer? Create layer with name meta-mysbcs and give priority 7.

ans: A layer is a logical collection of related recipes.

2.What is an image?create custom image with name mysbcs and inherit this image from core-image-minimal.

Ans:

Image is a top level recipe. (It inherits an image.bbclass).

inherit core-image

Which tells us that the definition of what actually gets installed is defined in the core-image.bbclass.

$ vim source/poky/meta/classes/core-image.bbclass

Image recipes set IMAGE_INSTALL to specify the packages to install into an image through image.bbclass.

Create an images directory

$ mkdir recipes-examples/images

Create the image recipe

$ vi recipes-examples/images/phy-image.bb

3.How to add usbutils,debug-tweaks,read-only-rootfs,splash in core-image-minimal.

Ans: vi conf/local.conf

IMAGE_FEATURES = "ssh-server-dropbear debug-tweaks read-only-rootfs splash"

IMAGE_INSTALL +=”usbutils”

4.Add gcc feature in core-image-minimal and then build the image after building in runqemu emulator once you get image write a mycalculator program and compile using gcc.

Ans:

tools-debug: Installs debugging tools such as strace and gdb.

tools-sdk: Installs a full SDK that runs on the device.

---->so to add splash feature in recipe

```
        IMAGE_FEATURES = "ssh-server-dropbear debug-tweaks read-only-rootfs splash
tools-debug tools-sdk"
```

5.What is recipe in yocto? What information present in recipe.

Ans:

Recipes are fundamental components in the Yocto Project environment.

Information such as:

Location from which to download the unaltered source

any patches to be applied to that source (if needed)

special configuration options to apply

how to compile the source files and

how to package the compiled output

6.What is the use of bitbake tool in yocto.what are the basic task user usually define in recipe.

Ans:

It basically performs the same functionality as of make.

It's a task scheduler that parses python and shell script mixed code

The code parsed generates and runs tasks, which are basically a set of steps ordered according to code's dependencies.

It reads recipes and follows them by fetching packages, building them and incorporating the results into bootable images.

It keeps track of all tasks being processed in order to ensure completion, maximizing the use of processing resources to reduce build time and being predictable.

7.How many stages are there to build any recipe.explain each of them.

Ans:

7 stages

Stage 1: Fetching Code (do_fetch)

The first thing your recipe must do is specify how to fetch the source files.

Stage 2: Unpacking (do_unpack)

All local files found in SRC_URI are copied into the recipe's working directory, in $BUILDDIR/tmp/work/

When extracting a tarball,BitBake expects to find the extracted files in a directory named <application>-<version>. This is controlled by the S variable.

Stage 3: Patching Code (do_patch)

Sometimes it is necessary to patch code after it has been fetched.

Any files mentioned in SRC_URI whose names end in .patch or .diff or compressed versions of these suffixes (e.g. diff.gz) are treated as patches

The do_patch task automatically applies these patches.

Stage 4: Configuration (do_configure)

Most software provides some means of setting build-time configuration options before compilation

Typically, setting these options is accomplished by running a configure script with options, or by modifying a build configuration file

Stage 5: Compilation (do_compile)

do_compile task happens after source is fetched, unpacked, and configured.

Stage 6: Installation (do_install)

After compilation completes, BitBake executes the do_install task

During do_install, the task copies the built files along with their hierarchy to locations that would mirror their locations on the target device.

Stage 7: Packaging (do_package)

The do_package task splits the files produced by the recipe into logical components.

Even software that produces a single binary might still have debug symbols, documentation, and other logical components that should be split out.

The do_package task ensures that files are split up and packaged correctly.

8.What if i have make file which don't have cleanTarget.

Ans:

CLEANBROKEN = "1"

If set to "1" within a recipe, CLEANBROKEN specifies that the make clean command does not work for the software being built.

Consequently, the OpenEmbedded build system will not try to run make clean during the do_configure task, which is the default behavior.

9.What should we add in recipe in do_install. how to add a package in libdir of any recipe.

Ans:

```
do_install() {
        install -d ${D}${bindir}
        install -m 0755 hello ${D}${bindir}
}
do_install_append() {
    install -d ${D}${libdir}
    install -m 0755 ${S}/lib/libmylibrary.so ${D}${libdir}/
}
```

10.What are the build time and run time dependence.

Ans:

Build Time Dependencies:

Your software uses a particular library. For example pthread/openssl.

Application cannot build without pthread/openssl library

Run Time Dependencies:

If your software internally calls a particular command for example (lspci)

If your software uses run time loading of dynamic library (dlopen)

Application/Software can build, but need them during execution

11.What are the steps to compile any program using autotools.

Ans:

Step1 : Configuration (./configure)

The configure script is responsible for getting ready to build the software on your specific system

It makes sure all of the dependencies for the rest of the build and install process are available

programs often written in C, usually need a C compiler to build them.

In these cases the configure script will establish that your system does indeed have a C compiler, and find out what it’s called and where to find it.

Step2 : Build (make)

Once configure has done its job, we can invoke make to build the software.

This runs a series of tasks defined in a Makefile to build the finished program from its source code.

Makefile comes with a template called Makefile.in and the configure script produces a customised Makefile specific to your system.

Step3 : Install (make install)

make install command will copy the built program, and its libraries and documentation, to the correct locations.

program’s binary will be copied to a directory on your PATH

program’s manual page will be copied to a directory on your MANPATH

12.What is the use of .bbappend file. what are the benefits of bbappend.

Ans:

.bbappend file in Yocto Project is used to extend or modify existing recipe files. Its benefits include maintainability and version compatibility.

13.How does bitbake picks up any recipe, what happens when you say bitbake<target>.

Ans:

When a recipe uses PROVIDES, that recipe's functionality can be found under an alternative name or names other than the implicit PN name

when PROVIDES reachs the target recipe then bit bake will picks the recipe

What happens when you say bitbake <target>?

1. Bitbake parses all the recipes

2. Bitbake looks through the PROVIDES list for each of the recipe

3. When the target matches PROVIDES list, it will build that recipe

14.What is virtual target?. What is the use of PREFERD_PROVIDER variable.

Ans:

A virtual target is a name that corresponds to some particular functionality.

The PREFERRED_VERSION variable supports limited wildcard use through the "%" character.

You can use the character to match any number of characters, which can be useful whe

15.Write a steps to update your kernel from old version to latest version.how to check which kernel vesion we are using in command line.

Ans:

```
$ uname -a
$ uname -r
```