# DeviceTree vtechstudy : Lesson_3

Q.1) DTBs are not linked with code, so the concept of initializing a pointer variable isn't valid.

What is a similar construct that is supported by DTBs?

A) node-linkage

B) property-envelope

C) #address-cell

D) phandle

Q.2) 1) **uart-device1 = <&qupv3_0>;** AND 2) **uart-device2 = &qupv3_0;** are both valid property statements in a .dts file and they refer to the same node. However, the syntax used is different and the value stored in the DTB is very different for each property.

Match each property with its correct definition:

A) phandle reference stored as a unique uint32_t

B) a uint32_t that is the node offset of the referenced node

C) a string that represents full path of the referenced node

D) a string that represents partial path of node, starting with parent

E) a uint32_t that is hash of node for optimized search/lookup

1) =

2) =

In Lesson_3/ folder, run make. Running the command **./lesson_3 -f tlmm.dtb** yields this output:

```
./lesson_3 running...
[main] read dtb_blob[tlmm.dtb]..size[48878]
Now do lookups using node names
Now do lookups using phandles
example() returned [FDT_ERR_QC_NOERROR]...
```

Add -t flag to end of above command-line to add tracing information.

Unlike in previous lessons, the -t flag isn't providing additional information on the console. If you review the code you will see that there should be output being generated by the -t flag. The key to why this is so, is the code on lines 73-80.

Q.3) What do you think this code is doing?

A) Reading the config node in the DTB

B) Setting the config node from the dtb_config variable

C) Reading the config property in the DTB

D) Setting the config property from the dtb_config variable

Running these simple examples from a command-line allows parameters to be specified on the command-line to control how the program runs. In these examples the parameters are controlling additional information being printed on the console. Consider that an embedded system, e.g. a boot-loader, has no way to read parameters from the command-line. The ability to parameterize or provide configuration details cannot come from the command-line, another mechanism is required.

Open QDTE and load the tlmm.dtb file that was created earlier in this lesson.  Find the config line and note there are three values: 0 0 1.  Edit the second value to be 1 and save the file.  Re-run the example and you should now see this output:

```
./lesson_3 running...

[main] read dtb_blob[tlmm.dtb]..size[48878]

example_3 running...


Now do lookups using node names

usec[1] to lookup [/tlmm/qup_l0_13] @offset[124]..phandle[2]

usec[63] to lookup [/tlmm/phase_flag_status_loc_16] @offset[10452]..phandle[173]

usec[166] to lookup [/tlmm/qdss_gpio_tracedata_loca_15] @offset[26312]..phandle[422]


Now do lookups using phandles

usec[2] to lookup [/tlmm/qup_l0_13] @offset[124]..phandle[2]

usec[118] to lookup [/tlmm/phase_flag_status_loc_16] @offset[10452]..phandle[173]

usec[308] to lookup [/tlmm/qdss_gpio_tracedata_loca_15] @offset[26312]..phandle[422]


example() returned [FDT_ERR_QC_NOERROR]...
```

Run the example a couple of times and confirm the usec output is similar to the above output.  By similar, the second grouping should have larger values in the usec box.

The three lookups represent nodes at the beginning, middle and end of the DTB file.

Q.4) Select all of the true statements:

A) lookups are O(1) complexity

B) string length of name being looked up is primary contributor to performance delta

C) lookups from the root node are faster than lookups from sub-nodes

D) node name lookups are faster than phandle lookups

In a text editor, open the file pinctrl.dtsi and compare the source code with the DTB in QDTE. One of the differences you should notice is the qup nodes in the dtsi file each has a single property, mux. The same nodes in the DTB file each have a second property named phandle. Somehow dtc is adding these properties.

Q.5) What is required for a phandle to be able to be generated for a node?

A) a unit-name

B) an ampersand (&)

C) a label

D) node name in angle brackets

Having reviewed the code in the FdtLib for the phandle calls, I believe there is a simple explanation for the bad performance of the phandle lookup compared to the name lookup. If you remember from Lesson_2, the name of the node is immediately after the tag that identifies the node structure in the file, so once the node is located, the name can be compared directly. The phandle is added by dtc as a new property to the node and as QDTE shows, the phandle property is added at the end of the node. Once the node is located it has to be scanned for a phandle property and if found the property value has to be extracted from the DTB and compared with the phandle the client is looking for.

Phandles are required to link nodes together within the tree, e.g. a device with an interrupt probably needs to be linked with the interrupt controller. Or an I2C device probably needs to be linked with the I2C bus controller it sits under, in the topography of the hardware. This linkage comes with a stiff performance penalty, as we have discovered.

Use this command-line to generate a second .dtb file we will use in this lesson**: ../../../../gendtb.py -f tlmm-broken.dts**

Note: gendtb.py is a simple script that combines running the C pre-processor on the .dts file and then using dtc on the generated C pre-processor output file. This mimics how the Linux kernel uses dtc and allows the use of simple #define macros as well as the #include mechanism. The output of the C pre-processor has a .pp file extension and has some value in reviewing it.

Running the command **./lesson_3 -f tlmm-broken.dtb** yields this output:

```
./lesson_3 running...

[main] read dtb_blob[tlmm-broken.dtb]..size[48647]

example_3 running...


Now do lookups using node names


fdt_getprop returned [(nil)]..len[-1]

get_prop_value[phandle] returned error[-1]


example() returned [FDT_ERR_NOTFOUND]...
```

Enabling the trace flag in config property doesn't change the output at all.

Q.6) Can you fix this problem? List the file-name, line number and corrected line below for each fix you need to apply.