Interview Questions Topic wise

he following questions are asked for me for Audio sw engineer (3-8years)exp.

1. About previous Audio Experience in various domains in Audio

2. Audio codecs which I have worked, Implementation and testing

3. Platforms which I have worked like intel arm cortex m4 Implementation

4. Previous projects and tools which I have used like gdb , Ide's Audacity etc

5. Audio Sw architecture flow.

6. audio policy manager working

7. audio flinger service working

8. dsp basics fir iir filter , sample rate conversions

9. codec optimizations techniques , memory constraints, stack memory utilization

10.mutex semaphore live situations,scheduling

11. adb log analysis , debugging tools like gdb trace 32.

13 .various services involved during playing song from media player to speaker

14. Alsa driver architecture and flow.

Aim : To integrate new Audio codec device to platform

Interview questions asked ,

basic question tell me abt your self

audio projects which i have worked

how codec optimization taken place

1.what are steps followed to integrate new device to kernel

* 1) it easier to add newly developed devices to existing systems, and (2) the development of entirely new types of devices, for which the existing standard interfaces are not always easy to apply.
* ***Device drivers*** are modules that can be plugged into an OS to handle a particular device or category of similar devices.
* Most of the devies cahracterized as block,character and network.

2. what is device tree and what consists of and its functionality

> Device tree is a [data structure](https://en.wikipedia.org/wiki/Data_structure) describing the hardware components of a particular computer so that the [operating system](https://en.wikipedia.org/wiki/Operating_system)'s [kernel](https://en.wikipedia.org/wiki/Kernel_(operating_system)) can use and manage those components, including the [CPU](https://en.wikipedia.org/wiki/Central_processing_unit) , the [memory](https://en.wikipedia.org/wiki/Computer_memory), the [buses](https://en.wikipedia.org/wiki/Bus_(computing)) and the [peripherals](https://en.wikipedia.org/wiki/Peripheral).

> They are located at arch/arm/boot/dts/

3. how we will set gpio pins and where we will set

> To manage the GPIO registration and allocation there is a framework inside the Linux kernel called **gpiolib**. This framework provides an API to both device drivers running in kernel space and user space applications.

> oldway is --- sysfs interface,  the interface to manage GPIO lines in user space [has always been in sysfs](https://www.kernel.org/doc/html/latest/driver-api/gpio/legacy.html#paths-in-sysfs) via files exported at /sys/class/gpio.

> newway is --- chardev interface , now we have a new API based on character devices to access GPIO lines from user space.

> Every GPIO controller (**gpiochip**) will have a character device in /dev and we can use file operations (open(), read(), write(), ioctl(), poll(), close()) to manage and interact with GPIO lines:

4. what probe in kernel where it will set

> The driver's init function calls pci\_register\_driver() which gives the kernel a list of devices it is able to service, along with a pointer to the probe() function. The kernel then calls the driver's probe() function once for each device.

This probe function starts the per-device initialization: initializing hardware, allocating resources, and registering the device with the kernel as a block or network device or whatever it is.

That makes it easier for device drivers, because they never need to search for devices or worry about finding a device that was hot-plugged. The kernel handles that part and notifies the right driver when it has a device for you to handle.

5. machine driver functionality

6. where the configuration of new device set.

Configuration of device set during driver initialization.

7. name the codec present in 8909/8905 chipsets

8. role of dai

9.what is interrupt , how it is generated and how we identify interrupt is generated .

> An **interrupt** is a signal sent to the processor that **interrupts** the current process. It may be **generated** by a hardware device or a software program. A hardware **interrupt** is often **created** by an input device such as a mouse or keyboard.

10.what is volatile variable and constant volatile

> const volatile' variable cannot be modified programmatically but can be modified by hardware.

> volatile will tell the compiler not to optimise code related the variable

11.what happens when interrupt generted how to handle

When an **interrupt occurs**, it causes the CPU to stop executing the current program. The control then passes to a special piece of code called an **Interrupt** Handler or **Interrupt** Service Routine.

Aim : To integrate new Audio codec device to platform

Interview questions asked ,

1.what are steps followed to integrate new device to kernel

2. what is device tree and what consists of and its functionality

3. how we will set gpio pins and where we will set

4. what probe in kernel where it will set

5. machine driver functionality

6.what is interrupt , how it is generated and how we identify interrupt is generated .

7.what is volatile variable

--> How you will start board bring-up for any given board? explain all steps...?

Board bring up activity is inclusive of assembly, hardware, firmware, software elements.

1)At first we need to check board has assembled properly in terms of soldering or any breakage during shipping.

2)Basic hardware test like individual devices and buses/interconnects are operational(ex: power up board and test basic functionality)

3)Finally systems hardware is loaded

Being an embedded developer we should know our board in detail.

Hardware - processor

types of memorys

i/o ports, switches

software - Which OS is supported like android/linux

Startup sequence:-

processor / controller start up - processor jumps to pre-designated address to run pre-programmed code( code looks for stage1 bootloader)

bootloader -

stage1 bootloader

IPL - Intial program loader(constrained to limited space)

stage2 bootloader(U-Boot or secondary bootloader)

loads kernel and passes arguments to kernel

Jumps to uncompress kernel if compressed

configure kernel based on arguments provide in bootloader

operating system start up

Jumps to uncompress kernel if compressed

configure kernel based on arguments provide in bootloader

setup kernel space

start first application init.rc

Application start up

Android OS booting happens from here ( system.img, userdata.img, vendor.img )

start various daemons

https://www.slideshare.net/anil\_pugalia/board-bringup-4566964 - collected from this link

--> How secure boot works, explain each steps of booting process... ??

The process of Secure Boot is where your OS boot images and code are authenticated against the hardware before they’re allowed to be used in the actual boot process

The hardware is set up beforehand in such a way that it only authenticates code generated using security credentials you trust.

it ensures that the boot and OS software is the intended manufacturer version and hasn’t been tampered with by any malicious party or process.

Android phone Secure Boot would restrict end users from running custom ROMs,

The Secure Boot process starts with a secret key, which is used to verify that the boot code is valid

https://www.electronicdesign.com/technologies/embedded-revolution/article/21806085/secure-boot-what-you-need-to-know

https://source.android.com/security/verifiedboot/verified-boot

Verified boot requires cryptographically verifying all executable code and data that is part of the Android version being booted before it is used. This includes the kernel (loaded from the boot partition), the device tree (loaded from the dtbo partition), system partition, vendor partition, and so on.

Hi all,

1)Explain Android architecture?

2)Explain code for Audio playback scenario?

3)Explain state diagram of media player object?

4)Role of awesome player in stagefright?

5)Functionality of audio flinger?

6)Role of audio policy service and audio policy manager?

7)What are the native services that start from media server?

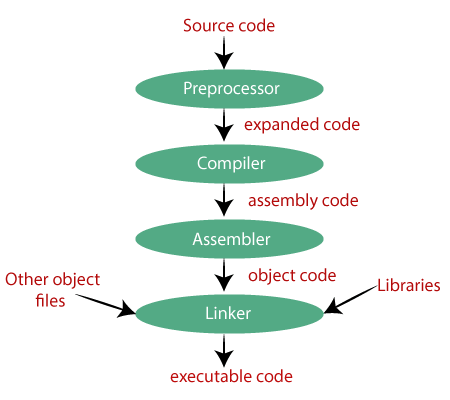
8)what is audio track and audio sink in context of playback?

Topics

-------

C programming ---

1.compilation process in c ?



2.what is diffrence b/w Assembly code,Object Code,Executable code ?

Assembly code : An **assembly language** is a low-level programming **language** designed for a specific type of processor. It may be produced by compiling source **code** from a high-level programming **language** (such as **C**/**C**++) but can also be written from scratch. **Assembly code** can be converted to machine **code** using an **assembler**

Object code :: **Object code** is a portion of machine **code** that has not yet been linked into a complete program. It is the machine **code** for one particular library or module that will make up the completed product.

Executable code :: **Executable code** generally refers to machine language.

3.what is cross compiler ? explaine b/w native compilation cross compilation ?

**1. Native Compiler :**  
Native compiler are compilers that generates code for the same Platform on which it runs. It converts high language into computer’s native language. For example Turbo C or GCC compiler

**2. Cross compiler :**  
A Cross compiler is a compiler that generates executable code for a platform other than one on which the compiler is running. For example a compiler that running on Linux/x86 box is building a program which will run on a separate Arduino/ARM.

4.what is c library functions ? diffrent types of user defined c library ? how they compiled and linked ?

* > Library functions in C language are inbuilt functions which are grouped together and placed in a common place called library.
* We can make use of these library functions to get the pre-defined output instead of writing our own code to get those outputs.
* Actually, function declaration, definition for macros are given in all header files.
* We are including these header files in our C program using “#include<file\_name.h>” command to make use of the functions those are declared in the header files.

5.Memory Layout of C program ?

A typical memory representation of C program consists of following sections.

1. Text segment  
2. Initialized data segment  
3. Uninitialized data segment  
4. Stack  
5. Heap

6.c storage classes ? explaine with an each, stress more on static.(life time, scope, initialization)

7.what is an os explain with a block diagram ---OR --- architecture of linux operating system

8.what is a process ? thread ? how they communicate ? diffrent types of IPC's

9.Diffrence b/w semaphore and mutex ?

10.what is spinnlock?

11.what is interept and what hapen when interept occres ..

12.what is VFS?

13.what is

9. write a C function to toggle a bit at nthy position ---> newNum = num ^ (1 << n);

10. write a C function to get nth bit of a number ----->bitStatus = (num >> n) & 1;

11.write a C function to set nth bit of a number ---->newNum = (1 << n) | num;

12.write a program to reverse a string

13.write a program to reverse a number

14. swap num using pointers, without temp variable

15. reverse a linked list

16. Find out a loop in a linked list

17.How to allocate memory dynamically for 2X2 matrix.

18.difference b/w structure and union, typedef, #define, enum

19. More programs on pointers(null pointer, Dangling pointer, Void pointer, function pointer)

20. Git commands

Introduce yourself

Previous company projects

current company projects and roles

Based on your resume it seems to developer ,how could you justice youself that you could fit for intergration

linux boot process

android boot process

Explain about i2c and uart

how will you debug code in gdb

if there there 5000 lines of code how you identify which error and how to set breakpoint

git commands with live situations

diffference between semaphore and mutex /

how to identfy particular node in multi node linked list if contain 500 nodes

how to pass arrays as function parameters

how to write c program without using semicolunm

difference between svn and git

what is git ?, why it needed

where source codes are preseved

how to sync source code

if developer deletes the code how will you who deleted using git

if our system is crashed and lost source code how to regenerate it , which command

Interview questions:

1).Compilation stages in c

2).Palindrome function

3).Storage classes

4).Memory layout

5).Find the substring in mainstring by using userdefined function.

6).Swap the two variables by using bitwise operators

7).Volatile keyword

8) .git commands like git cherrypick, git rebase, git merge.

9). How to avoid merge conflict

10).QXDM,QPST, QFIL tools

11) .Build procedure

12).Difference between the hlos and nonhlos.

13). Fastboot commands.

14). If you get any error in souce code after building ,what would you do .

15).Without including headerfile of printf and scanf, whether the program gets compiled or not.

16).Difference between the list and tuple

17). How to change the tuple variable in python.

18).Roles and responsibities of project.

19).Project explanantion in step by step.

20).Difference between the tcpip and udp protocol

21).Udp header file.

22).Testing scenario of wavfiles .

24).Kernel cross compiling procedure.

23).Linux commands.

**jio interview questions**

How you will start board bring-up for any given board? explain all steps...

--> Let you have particular Android OS version, how you will upgrade, what are issues you faced?

--> In Qualcomm what are all devices that you have worked. what is your job role, what type of issues you have solved?

--> Explain each bootup steps, what happened in each steps...

--> Let after bootloader initialized it is not able to detect/find kernel address... how you will debug/fix this?

--> Let Primary bootloader is not able to initialize a driver, how you will debug/fix this?

--> How secure boot works, explain each steps of booting process...

--> What you have worked in Multimedia Graphics, what type of issues you have solved, how you are analyzing source code , code flow to find root cause of issue...

--> what is difference between Semaphore, Mutex, Spinlock...?

---> What is Deadlock , how you will avoid deadlock ?

---> What are IPC mechanism you know...?

---> Difference between Kmalloc and vmalloc?

Interview Questions

1 . What is the storage class in c

2 . What is the dynamic memory allocation .

3 . What is the void pointer

4 . C code compailation stages .

5 . Explain in brife about the data stractures

6 . How to add node at end and middle in single linked list.

7 . What is the difference between the arrays and linked list.

8 . What is the command to change the permision to files or directory.

9 . What is the command to know the list file in directory .

10 . How to know running processor in you are pc.

11 . How to create a thread.

12 . What is the life cycle of thread .

13 . What is the difference between thread and process

14 . What is the use of git expalin in brife.

15 . what is the shell.

16 . what are kernal version control tools .

17 . what is linux version you are using.

18 . what is command know only sud - directries present in a directory.

Qualcomm Interview Question :

1. Explain the Booting process.

2. Explain I2C communication protocol

3. write c program swap two number in three methods

4. write c program to add two number using function

5. Explain inline function in c.

6. what is the macro?

7. what is the structure? Give the examples

8. what is the difference between macro and inline function?

9. give the examples of structure and find the sizeof structure.

10. what is the structure padding? And how to resolve it ?

11. what is the pragama pack and bitfield?

12. Explain Projects detail.

13. different between process and thread?

**Build and Integration interview quesions**

First Round:

1.Roles and responsibities of project.

2.Project explanantion in step bt step.

3.Git commands like git pull,fetch,merge,rebase,reset (Described a situation and asked which command is suitable).

4.Linux commands--->Given few examples and asked which command is used there.(chown,grep,rm)

5.How much time took for syncing a source code?

6.Build procedure

7.fastboot commands.

8.QPST,QFIL tools.

Second Round:

1.Git pull and fetch differences.

2.Reverse a linked list.

3.Storage classes.

4.Differences between static and extern.

5.Set,list,tuple,dictionary in python.

6.fastboot commands

7.QXDM log analysis(basic rrc messages)

8.How can we connect a QRD and flash it other than Fastboot?

9.If wifi is not working how will u figure out(if developers are absent and an urgent requirement for a release)?