

Demand paging :-

<http://www.geeksforgeeks.org/virtual-memory-operating-systems/>

pointers :-

<http://www.geeksforgeeks.org/dangling-void-null-wild-pointers/>

strings :-

<http://www.geeksforgeeks.org/storage-for-strings-in-c/>

function pointer :-

<http://www.geeksforgeeks.org/function-pointer-in-c/>

linux-device- drivers :-

<http://www.opersys.com/downloads/cc-slides/linux-device-drivers/linux-device-drivers-120203.pdf>

5/9/17

1.adb debugging tool

<https://developer.android.com/studio/command-line/adb.html>

2.Linux device driver /linux kernel

<http://learnlinuxconcepts.blogspot.in/2014/03/why-kernel-code-running-in-interrupt.html>

2.1 Use above link or bellow - for interrupts topic.

<http://learnlinuxconcepts.blogspot.in/2014/02/interrupts.html>

spinlock

<http://www.albahari.com/threading/part5.aspx# SpinLock and SpinWait>

spinlock and read write lock

<http://locklessinc.com/articles/locks/>

poling vs spinlock

<https://softwareengineering.stackexchange.com/questions/301947/how-is-spinlock-different-from-polling>

spinlock

<https://en.wikipedia.org/wiki/Spinlock>

06/09/2017

Touch Screen

<http://matthewhalpern.com/publications/mosaic-ispass-2015.pdf>

core,chipset for phones

http://www.gsmarena.com/lg_ray-7773.php

Input Output system calls

<http://www.geeksforgeeks.org/input-output-system-calls-c-create-open-close-read-write/>

Touch screen

<http://papers.www2017.com.au.s3-website-ap-southeast-2.amazonaws.com/proceedings/p331.pdf>

Trace32

<http://www.lauterbach.com/tutorial.pdf>

07/09/2017

Device drivers infrastructure

The Basic Device Driver-Model Structures

<https://01.org/linuxgraphics/gfx-docs/drm/driver-api/infrastructure.html>

* Device Power Management Basics

<https://01.org/linuxgraphics/gfx-docs/drm/driver-api/pm/devices.html#two-models-for-device-power-management>

** Device Power Management Operations :-

<http://elixir.free-electrons.com/linux/latest/source/include/linux/pm.h>

Structure in Functions:-

<https://www.programiz.com/c-programming/c-structure-function>

i2c protocol

<https://01.org/linuxgraphics/gfx-docs/drm/driver-api/i2c.html>

<http://www.firmcodes.com/i2c-protocol/>

c

<http://www.c4learn.com/c-programs/how-to-create-your-own-header-file-in-c.html>

08/09/2017

*****Touch Screen

[https://books.google.co.in/books?](https://books.google.co.in/books?id=JxyB_p1g6uMC&pg=PA195&lpg=PA195&dq=android+touch+driver+flow&source=bl&ots=CHaydDxSXQ&sig=HwplR3mfBGK9MYR8eorf3Td89yE&hl=en&sa=X&ved=0ahUKEwiI5t347JTWAhWKp48KHQVoCIUQ6AEIVTAH#v=onepage&q=android%20touch%20driver%20flow&f=false)

[id=JxyB_p1g6uMC&pg=PA195&lpg=PA195&dq=android+touch+driver+flow&source=bl&ots=CHaydDxSXQ&sig=HwplR3mfBGK9MYR8eorf3Td89yE&hl=en&sa=X&ved=0ahUKEwiI5t347JTWAhWKp48KHQVoCIUQ6AEIVTAH#v=onepage&q=android%20touch%20driver%20flow&f=false](https://books.google.co.in/books?id=JxyB_p1g6uMC&pg=PA195&lpg=PA195&dq=android+touch+driver+flow&source=bl&ots=CHaydDxSXQ&sig=HwplR3mfBGK9MYR8eorf3Td89yE&hl=en&sa=X&ved=0ahUKEwiI5t347JTWAhWKp48KHQVoCIUQ6AEIVTAH#v=onepage&q=android%20touch%20driver%20flow&f=false)

11/09/17

i2c protocol

<http://elixir.free-electrons.com/linux/v2.6.39/source/Documentation/i2c/i2c-protocol>

Power management intel

http://download.intel.com/support/motherboards/server/sb/power_management_of_intel_architecture_servers.pdf

<https://software.intel.com/en-us/articles/power-management-states-p-states-c-states-and-package-c-states>

<https://people.cs.pitt.edu/~kirk/cs3150spring2010/ShiminChen.pptx>

https://events.linuxfoundation.org/sites/events/files/slides/kernel_PM_plain.pdf

<https://www.kernel.org/doc/ols/2012/ols2012-mansoor.pdf>

12/09/17

spinlock , semaphore, mutex

<https://unix.stackexchange.com/questions/5107/why-are-spin-locks-good-choices-in-linux-kernel-design-instead-of-something-more>

C

<http://cinterviewquestionandanswer.blogspot.in/2014/01/memory-layout-of-c-programs.html>

13/09/17

IMP Links

http://processors.wiki.ti.com/index.php/TI-Android-GingerBread-2.3.4-DevKit-2.1_PortingGuides#Touchscreen

<http://ww1.microchip.com/downloads/en/DeviceDoc/AR1020-AR1021-LINUX-SPI-I2C-V102.pdf>

https://android.googlesource.com/kernel/bcm/+23d376ef33aa4c500a5ea24a290f029d5f8e2de3/drivers/input/touchscreen/egalax_i2c_ts.c

<https://linux.die.net/man/1/objdump>

https://android.googlesource.com/kernel/bcm/+55040d0cfa091003fb7840fd057e109969ac2440/drivers/input/touchscreen/synaptics_i2c_rmi.c

i2c

<https://android.googlesource.com/kernel/bcm/+android-bcm-tetra-3.10-marshmallow-mr1-wear-release/drivers/i2c/i2c-core.c>

<http://ww1.microchip.com/downloads/en/DeviceDoc/AR1020-AR1021-LINUX-SPI-I2C-V102.pdf>

touch -synptic

http://www.aotom.com/Upload/%E4%BA%A7%E5%93%81%E4%B8%AD%E5%BF%83/%E6%96%B0%E6%80%9D/S7817Datasheet50500057001Rev1_S7817-15365878520.pdf

15/09/17

Q.what is NCURSES?

NCURSES is a clone of the original System V Release 4.0 (SVr4) curses. It is a freely distributable library, fully compatible with older version of curses. In short, it is a library of functions that manages an application's display on character-cell terminals. In the remainder of the document, the terms curses and ncurses are used interchangeably.

What we can do with NCURSES

NCURSES not only creates a wrapper over terminal capabilities, but also gives a robust framework to create nice looking UI (User Interface)s in text mode. It provides functions to create windows etc. Its sister libraries panel, menu and form provide an extension to the basic curses library. These libraries usually come along with curses. One can create applications that contain multiple windows, menus, panels and forms. Windows can be managed independently, can provide 'scrollability' and even can be hidden.

Menus provide the user with an easy command selection option. Forms allow the creation of easy-to-use data entry and display windows. Panels extend the capabilities of ncurses to deal with overlapping and stacked windows.

Interrupts

<http://www.cs.toronto.edu/~demke/469F.06/Lectures/Lecture6.pdf>
<http://www.electronics.dit.ie/staff/tscarff/6800/Interrupts/interrupts.htm>

Q.Are there any reasons why I shouldn't call the printf function from within an interrupt?

ANSWER

There are numerous good reasons to avoid calling printf from an interrupt routine.

Interrupt routines are typically used to respond to events that occur quickly. For this reason, it is usually important that interrupt service routines run fast. The printf

function is not a fast routine. Especially if you consider that the output from printf may output to the slow serial port. At 1200 baud, a 100 character string takes almost 1 second to send. The printf function will therefore take more than that to execute. In this case, your interrupt service routine will take around 1 second to execute. If that interrupt happens faster than once per second, you will lose interrupts.

18/09/17

http://www.atmel.com/Images/DDI0029G_7TDMI_R3_trm.pdf

<http://www.arm.linux.org.uk/docs/kerncomp.php>

19/09/17

Writing Modules for Multiple Kernel Versions

<http://www.tldp.org/LDP/lkmpg/2.4/html/c577.htm>

<https://www.javatpoint.com/c-interview-questions>

20/09/17

<http://opensourceforu.com/2011/08/io-control-in-linux/>

<http://free-electrons.com/doc/books/lkn.pdf>

<http://free-electrons.com/doc/training/linux-kernel/linux-kernel-slides.pdf>

n

<http://www.ee.surrey.ac.uk/Teaching/Unix/>

* 21/09/17

<http://www.sanfoundry.com/c-programming-examples-strings/>

https://wiki.archlinux.org/index.php/Step-by-step_debugging_guide

****22/09/17**

Exporting symbols from module

<http://tuxthink.blogspot.in/2011/07/exporting-symbols-from-module.html>

<http://opensourceforu.com/2012/05/linux-device-drivers-module-interactions/>

cross compiling linux for ARM

<http://tuxthink.blogspot.in/2010/05/cross-compiling-linux-for-arm.html>

IOCTL

<http://tuxthink.blogspot.in/2011/01/creating-ioctl-command.html>

process

<http://www.informit.com/articles/article.aspx?p=368650&seqNum=2>

<http://www.geeksforgeeks.org/io-interface-interrupt-dma-mode/>

25/09/17

threads

<http://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html>

<http://www.thegeekstuff.com/2009/03/8-essential-vim-editor-navigation-fundamentals/>

26/09/17

clone() system call

<http://www.linuxjournal.com/article/5211>

GIT

<https://git-scm.com/book/en/v2>

GIT videos

<https://git-scm.com/doc>

<http://opensourceforu.com/2010/10/debugging-linux-kernel-with-debugfs/>

-

Setting up DebugFS

If you are using one of the latest distributions, chances are that debugfs is already set up on

your machine. If you're compiling the kernel from scratch, make sure you enable debugfs in the kernel configuration. Once you reboot to your newly compiled kernel, check if debugfs is already mounted, with the following command:

```
# mount | grep debugfs  
none on /sys/kernel/debug type debugfs (rw)
```

If you see output as above, you have debugfs pre-mounted. If not, you can mount it (as root) with the command shown below:

```
# mount -t debugfs nodev /sys/kernel/debug
```

If you want it to be available on every reboot, append an entry in `/etc/fstab` as follows:

```
debugfs /sys/kernel/debug debugfs defaults 0 0
```

Once mounted, you can view a lot of files and directories in `/sys/kernel/debug`, each belonging to one or the other subsystem.

Structure in c

<http://fresh2refresh.com/c-programming/c-struct-memory-allocation/>

task_struct()

<http://opensourceforu.com/2016/03/the-life-of-a-process/>

05/10/17

TCP/IP

<http://www.ggu.ac.in/download/Class-Note14/lesson13.02.14.pdf>

RTOS

<http://user.it.uu.se/~yi/courses/rts/dvp-rts-08/notes/RTOS.pdf>