For more information

There are numerous sources of information on Linux kernel development and related topics. First among those will always be the Documentation directory found in the kernel source distribution. The top-level :ref:`process/howto.rst <process_howto>` file is an important starting point; :ref:`process/submitting-patches.rst <submittingpatches>` and :ref:`process/submitting-drivers.rst <submittingdrivers>` are also something which all kernel developers should read. Many internal kernel APIs are documented using the kerneldoc mechanism; "make htmldocs" or "make pdfdocs" can be used to generate those documents in HTML or PDF format (though the version of TeX shipped by some distributions runs into internal limits and fails to process the documents properly).

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\process\[linux-master][Documentation][process]8.Conclusion.rst`, **line 6**); *backlink*
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\process\[linux-master][Documentation][process]8.Conclusion.rst`, **line 6**); *backlink*
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\process\[linux-master][Documentation][process]8.Conclusion.rst`, **line 6**); *backlink*
>
> Unknown interpreted text role "ref".

Various web sites discuss kernel development at all levels of detail. Your author would like to humbly suggest https://lwn.net/ as a source; information on many specific kernel topics can be found via the LWN kernel index at:

https://lwn.net/Kernel/Index/

Beyond that, a valuable resource for kernel developers is:

https://kernelnewbies.org/

And, of course, one should not forget https://kernel.org/, the definitive location for kernel release information.

There are a number of books on kernel development:

Linux Device Drivers, 3rd Edition (Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman). Online at https://lwn.net/Kernel/LDD3/.

Linux Kernel Development (Robert Love).

Understanding the Linux Kernel (Daniel Bovet and Marco Cesati).

All of these books suffer from a common fault, though: they tend to be somewhat obsolete by the time they hit the shelves, and they have been on the shelves for a while now. Still, there is quite a bit of good information to be found there.

Documentation for git can be found at:

https://www.kernel.org/pub/software/scm/git/docs/

https://www.kernel.org/pub/software/scm/git/docs/user-manual.html

Conclusion

Congratulations to anybody who has made it through this long-winded document. Hopefully it has provided a helpful understanding of how the Linux kernel is developed and how you can participate in that process.

In the end, it's the participation that matters. Any open source software project is no more than the sum of what its contributors put into it. The Linux kernel has progressed as quickly and as well as it has because it has been helped by an impressively large group of developers, all of whom are working to make it better. The kernel is a premier example of what can be done when thousands of people work together toward a common goal.

The kernel can always benefit from a larger developer base, though. There is always more work to do. But, just as importantly, most other participants in the Linux ecosystem can benefit through contributing to the kernel. Getting code into the mainline is the key to higher code quality, lower maintenance and distribution costs, a higher level of influence over the direction of kernel development, and

more. It is a situation where everybody involved wins. Fire up your editor and come join us; you will be more than welcome.