

# Kernel Engineer candidate test instructions:

This test exercise consists of three problems to solve. To proceed, you must solve them to the best of your ability and submit results for review. These exercises are real-world and are representative of our daily work.

Please tar up all of your response submissions into a single archive to submit for review.

## Q1 Code Review

Code review exercises are important because a significant amount of the Canonical kernel team's workload relies on being able to do quick but thorough code review for patches and bugs submitted. A solid base in code review skills are not only beneficial to kernel maintenance but also kernel software development.

Review the `snippit.c` and create a file with your answers to submit.

- a. Explain what the code is doing
- b. See if you spot any errors and enumerate them if you find one or more
- c. If you find any errors, how would you resolve them?

## Q2 Kernel Development Exercise

A smaller but vitally important part of our work is the ability to develop tight clean kernel code.

In this exercise you will:

- Create a kernel module that can be loaded and unloaded with `insmod/rmmod`
- The kernel module will create a `proc` entry and register a `read/write file_op`
- On each write to the `proc`, a copy of the buffer will be stored in a data structure
- Each read from the `proc` will fill the read buffer with list entries up to the read length (buffer size)
- List memory and `proc` entry will get properly cleaned up when module is unloaded

Develop your module against the Ubuntu Jammy 5.15 kernel to be found [here](#), from the **Ubuntu-5.15.0-83.92** tag.

You are encouraged to demonstrate your knowledge with additional detail not explicitly asked for above. In addition to submitting your code files please also include any files as you might submit to upstream Linux.

## Q3 Kernel Backporting and Git Usage

This question is to test your skills on Linux kernel backporting and git usage. You will be asked to backport to an Ubuntu kernel a patchset that was submitted to an external mailing list. Please follow the instructions below. For each step below, describe the actions you take and show your work. Even if you do not have sufficient time to complete the entire question, or are unable to finish the entire set of tasks, please turn in the work you have done.

1. The patchset that you need to backport is

<https://patchwork.kernel.org/project/linux-wireless/cover/20211024152037.332948-1-luca@coelho.fi/>, titled “[00/12] iwlwifi: updates intended for v5.16 2021-10-24 part 2”, which consists of these commits:

```
cbaa6aeedee5 iwlwifi: bump FW API to 67 for AX devices
af84ac579c66 iwlwifi: mvm: extend session protection on association
6905eb1c3b9e iwlwifi: rename CHANNEL_SWITCH_NOA_NOTIF to CHANNEL_SWITCH_START_NOTIF
cf7a7457a362 iwlwifi: mvm: remove session protection on disassoc
a6175a85ba33 iwlwifi: mvm: fix WGDS table print in iwl_mvm_chub_update_mcc()
523de6c872ca iwlwifi: rename GEO_TX_POWER_LIMIT to PER_CHAIN_LIMIT_OFFSET_CMD
4d4cbb9b8e56 iwlwifi: mvm: d3: use internal data representation
9da090cdbcfa iwlwifi: mvm: update RFI TLV
45fe1b6b6c99 iwlwifi: mvm: don't get address of mvm->fwrt just to dereference as a
pointer
698b166ed346 iwlwifi: mvm: read 6E enablement flags from DSM and pass to FW
1a5daead217c iwlwifi: yoyo: support for ROM usniffer
91000fdf8219 iwlwifi: fw: uefi: add missing include guards
```

2. Apply the patchset “[00/12] iwlwifi: updates intended for v5.16 2021-10-24 part 2” on top of the same Ubuntu-5.15.0-83.92 tree you used in Q2 above
3. You may need to cherry-pick some other commits for resolving any conflicts. The number of required commits to resolve the conflicts should be less than 10.
4. After the backport is done, verify the kernel builds successfully in the “Ubuntu way”, by following the steps listed below:
  1. Install Ubuntu 22.04 release on a machine or in a VM
  2. \$ sudo apt build-dep linux # to prepare build environment
  3. Change directory to the prepared kernel and build using:  
\$ fakeroot debian/rules clean binary-genericYou can find more information on how to build a Ubuntu kernel on [wiki.ubuntu.com](https://wiki.ubuntu.com).
5. Export the entire backport as a patch series and write down how you think when backporting commits and submit along with the answers to the other questions.

Please do not share these exercises.