

Free Page Reporting

Free page reporting is an API by which a device can register to receive lists of pages that are currently unused by the system. This is useful in the case of virtualization where a guest is then able to use this data to notify the hypervisor that it is no longer using certain pages in memory.

For the driver, typically a balloon driver, to use of this functionality it will allocate and initialize a `page_reporting_dev_info` structure. The field within the structure it will populate is the "report" function pointer used to process the scatterlist. It must also guarantee that it can handle at least `PAGE_REPORTING_CAPACITY` worth of scatterlist entries per call to the function. A call to `page_reporting_register` will register the page reporting interface with the reporting framework assuming no other page reporting devices are already registered.

Once registered the page reporting API will begin reporting batches of pages to the driver. The API will start reporting pages 2 seconds after the interface is registered and will continue to do so 2 seconds after any page of a sufficiently high order is freed.

Pages reported will be stored in the scatterlist passed to the reporting function with the final entry having the end bit set in entry `nent - 1`. While pages are being processed by the report function they will not be accessible to the allocator. Once the report function has been completed the pages will be returned to the free area from which they were obtained.

Prior to removing a driver that is making use of free page reporting it is necessary to call `page_reporting_unregister` to have the `page_reporting_dev_info` structure that is currently in use by free page reporting removed. Doing this will prevent further reports from being issued via the interface. If another driver or the same driver is registered it is possible for it to resume where the previous driver had left off in terms of reporting free pages.

Alexander Duyck, Dec 04, 2019