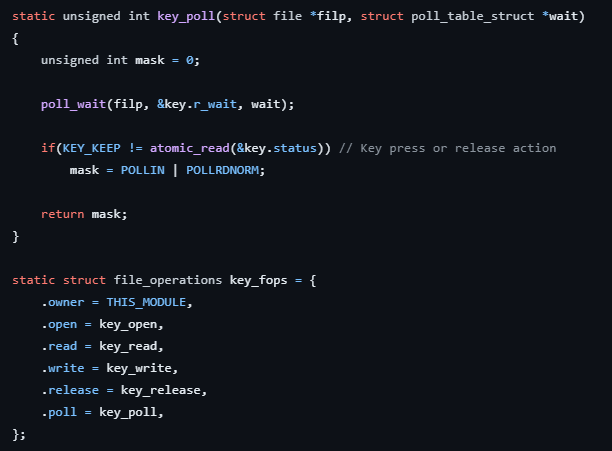
1. 專案15\_nonblockio:

* 與專案14\_blockio類似，本專案以Linux Kernel中提供的Poll函式來實現，非阻塞IO。非阻塞IO和阻塞IO最大的區別是當設備不可操作時進程不會休眠而是一直不斷輪詢，當輪詢時間超出預設時間時會給系統報錯。

1. 輪詢:

* 所謂輪詢就是，進程一直查看設備是否空閒可執行操作，所以CPU一直沒有閒著不斷查詢設備的IO模式。
* 在Linux系統中提供poll()函式用於處理輪詢。

1. nonblockio實作:
2. nonblockio相對blockio簡單，指需要再設備操作函式file\_operations中定義poll()函式即可。(如下圖一)至於timeout超時部分需要再應用城市中設定並定義其相應的處理。
3. 透過poll\_wait()函式將文件指標和等待隊列頭添加到輪詢表中，而POLLIN 表示有數據可讀，POLLRDNORM 表示普通數據可讀。



(圖一:file\_operations操作函式結構體)

※結論:

* + 本專案實現以poll()函數實現nonblockio功能。
* poll 函數的作用是：

1. 將當前進程加入到設備的等待隊列中，當有按鍵事件發生時喚醒該進程。
2. 檢查按鍵的狀態，如果有按鍵按下或釋放事件發生，設置相應的狀態標誌。
3. 返回狀態標誌給調用者，以通知用戶空間程序設備文件的當前狀態。

* 這種機制允許User Space中應用程序使用 poll或select系統調用來高效地監視設備文件的狀態，而無需不斷地輪詢設備文件，從而減少 CPU資源的占用。
  + 由於剩餘部分都與先前專案14\_blockio一樣，次裡就不贅述了。