

1. 如在呼叫 `recv` 時將 `flags` 參數設為 `MSG_PEEK`，且沒有額外設定其他 `flag` 值，則下列敘述何者正確？(複選) ①可以回傳 receiving queue 中可讀取的 byte 數目 ②不會真正將資料從 receiving queue 搬入 user buffer 中 ③如果 receiving queue 中沒有資料，則此呼叫會被 block ④如為 TCP socket，則此呼叫的傳回值與接下來呼叫 `recv` 的傳回值可能不同 (20%)

2. 如果要對 socket 使用標準的 C library function，下列敘述何者有誤？(單選) ①必須使用 `fileno` 將 FILE pointer 轉換為 socket file descriptor ②如要達成 socket 全雙工(full-duplex)的功能，可使用 `fdopen` 將同一 socket file descriptor 開啟兩次 ③每次呼叫 `fputs` 後使用 `fflush` 可避免標準 I/O 預設為 fully-buffered 所產生的問題 ④呼叫 `setvbuf` 可將標準 I/O 設定為 line-buffered (20%)

3. 下列哪些 I/O 函式可使用多個分散的(scattered) buffer？(複選) ①`read/write` ②`readv/writev` ③`recv/send` ④`recvfrom/sendto` ⑤`recvmsg/sendmsg` ⑥`fputs/fgets` (20%)

4. 考慮教科書中右邊這段程式碼。請問在★處發生何事？(複選) ①`recvfrom` 無法從 socket 收到任何資料 ②對方已關閉連線 ③`recvfrom` 被 SIGALARM 信號中斷 ④系統中已有其它 alarm 呼叫在執行中 ⑤`recvfrom` 被 SIGCHLD 信號中斷 ⑥`recvfrom` 被 SIGIO 信號中斷 ⑦`recvfrom` 執行超過時間尚未收到資料 (20%)

```
alarm(5);
if ( (n = recvfrom(...)) < 0 ) {
    if (errno == EINTR) //★ timeout
        fprintf(stderr, "XXX\n");
    else
        err_sys("recvfrom error");
} else {
    alarm(0);
    recvline[n] = 0;
    Fputs(recvline, stdout);
}
```

5. 下列何者無法防止對 socket 進行的 I/O 動作被 block forever？(單選) ①在 I/O 動作前呼叫 `alarm` ②設定 `SO_RCVTIMEO` 及 `SO_SNDTIMEO` socket options ③使用 `readv/writev` 取代 `read/write` ④呼叫 `select` 等待 I/O 裝置時設定 `timeval` 結構的內容並傳入其位址 ⑤使用 `recv/send` 設定 `flags` 參數為 `MSG_DONTWAIT` (20%)