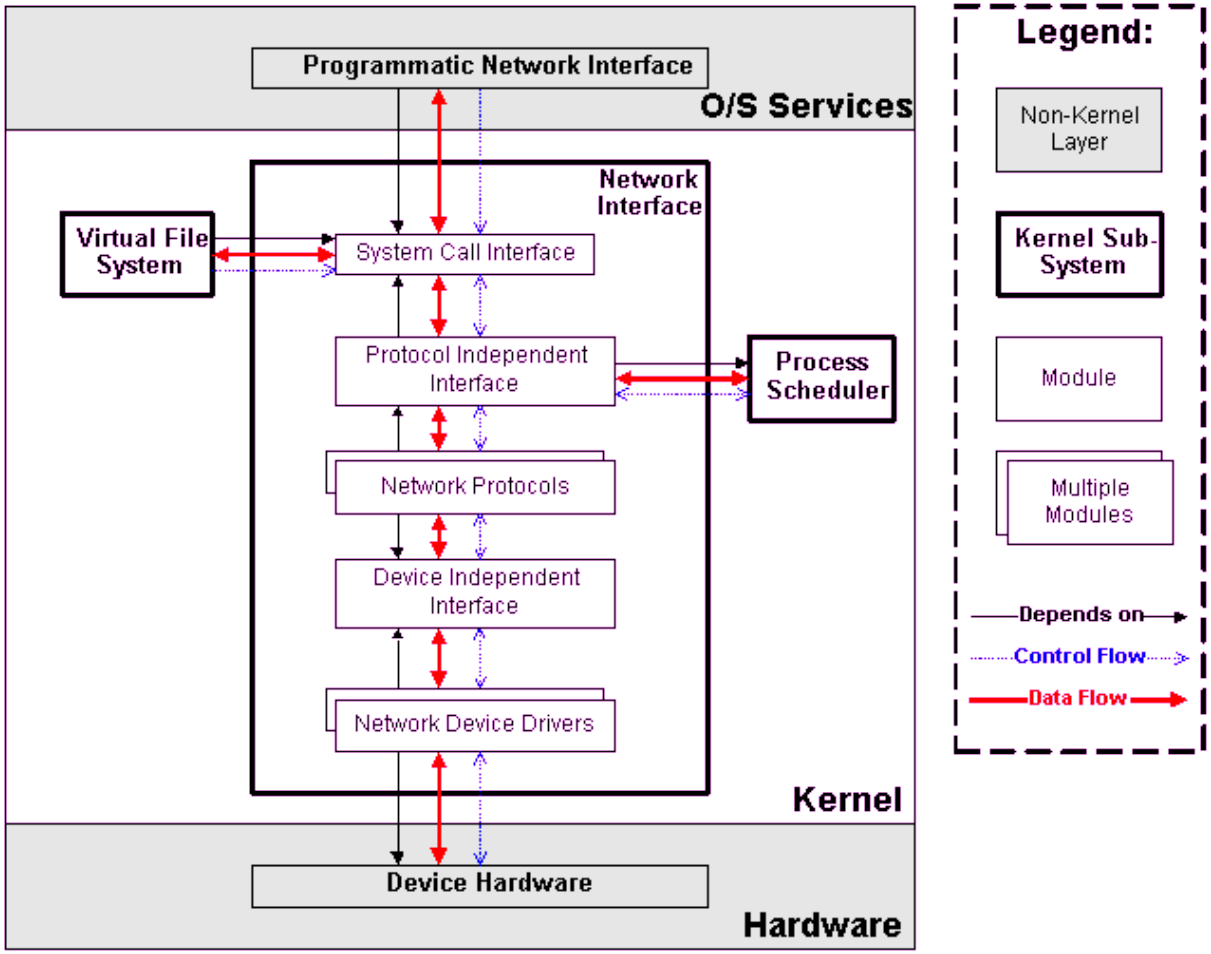
01.1. Призначення та структура мережної підсистеми ОС

- Назва: Мережна підсистема
  = Networking subsytem
  = Network Stack

- Приклад: мережна підсистема ОС Linux

- Структура мережної підсистеми ОС Linux, як і переважна більшість мережних підсистем сучасних ОС, в цілому відображає багаторівневу архітектуру стеку мережних протоколів.

01.2. Призначення та структура мережної підсистеми ОС



Структура мережної підсистеми ОС Linux

01.3. Призначення та структура мережної підсистеми ОС

Призначення:

1. Мережна підсистема дозволяє системі Linux з'єднуватись з іншими системами в комп'ютерній мережі.

2. Підримка різних типів мережних пристроїв (мережні адаптери Ethernet, Wi-Fi та ін.) та різних мережних протоколів, які використовуються для обміну даними.

01.4. Призначення та структура мережної підсистеми ОС

3. Мережна підсистема забезпечує абстрагування від низькорівневих деталей роботи як пристроїв, так і протоколів,

внаслідок чого користувацькі процеси та інші підсистеми ядра можуть користуватись мережою в стандартний уніфікований спосіб без обов'язкового уточнення який саме мережний пристрій та протокол застосовуються для передачі/прийому даних.

01.5. Призначення та структура мережної підсистеми ОС

Основні складові (модулі) мережної підсистеми:

1. Драйвери мережних пристроїв (Network device drivers):

- Забезпечують взаємодію з відповідними апаратними пристроями.
- Кожним наявним мережним пристроєм керує свій окремий драйвер.
- Часто використовується назва: NIC driver, де NIC - network interface controller/card.

01.6. Призначення та структура мережної підсистеми ОС

2. Модуль уніфікованого інтерфейсу мережних пристроїв (Device independent interface):

- Забезпечує стандартний інтерфейс доступу до апаратного забезпечення всіх мережних пристроїв,
- приховуючи від інших модулів мережної підсистеми низькорівневі деталі їхньої роботи.
- Приклад: приховування різниці між роботою Ethernet та Wi-Fi.

01.7. Призначення та структура мережної підсистеми ОС

3. Модулі мережних протоколів (Network protocols)

- Забезпечують логіку роботи стеку мережних протоколів за принципом багаторівневості (OSI model: Open Systems Interconnection model).

- Приклади стеків протоколів: TCP/IP (Internet), IPX/SPX (Novell), SNA (IBM).

01.8. Призначення та структура мережної підсистеми ОС

4. Модуль уніфікованого інтерфейсу стеку протоколів (Protocol independent interface):

- Забепечує стандартний інтерфейс доступу до стеку протоколів, незалежний від апаратних засобів та мережних протоколів.

- Цей модуль використовується іншими підсистемами ядра для доступу до мережі без конкретної прив'язки до протоколів і мережної апаратури.

01.9. Призначення та структура мережної підсистеми ОС

5. Модуль інтерфейсу системних викликів (System calls interface):

- Обмежує набір функцій мережної підсистеми, які можуть бути використані користувацькими процесами.
- Реалізує стандартний програмний інтерфейс для використання функцій мережної підсистеми.

01.9. Призначення та структура мережної підсистеми ОС

- В ОС Linux та більшості інших UNIX-подібних ОС використовується реалізація модулів (4) та (5) під назвою сокети Берклі (Berkeley sockets).
- Сокети Берклі — це бібліотека функцій та відповідний програмний інтерфейс транспортного рівня (в моделі OSI).
- Сокети Берклі забезпечують уніфікований інтерфейс до широкого набору протоколів (IP, UDP, TCP та ін.) та надають стандартний механізм організації обміну даними за схемою «клієнт-сервер».

01.10. Призначення та структура мережної підсистеми ОС

Основні функції мережної підсистеми ОС Linux:
1) надання програмного інтерфейсу транспортного рівня (сокети Берклі);
2) підтримка стеку мережних протоколів (TCP/IP, UDP/IP, IPX/SPX, AppleTalk та ін.);
3) маршрутизація (routing) та пересилання (forwarding) пакетів даних;
4) фільтрація пакетів даних (модуль Netfilter);
5) управління трафіком на мережному рівні (IP);
6) диспетчеризація пакетів даних (network scheduler);
7) надання абстракції мережних інтерфейсів (network interfaces).

01.11. Призначення та структура мережної підсистеми ОС

Напрямки розробки мережної підсистеми ОС Linux:

1. Розробка механізму «вбудованих» процедур обробки пакетів даних. Приклади: модуль Netfilter (фільтрація пакетів, NAT) та процедури управління мережним трафіком. Дані процедури передбачають наявність коду, роботу якого може конфігурувати користувач, у базовому процесі обробки пакетів даних.

2. Підвищення продуктивності програмних реалізацій логіки роботи протоколів (підвищення пропускної здатності, зменшення затримки). Приклад: вдосконалення алгоритмів управління навантаженням (congestion control) для протоколу TCP.

01.12. Призначення та структура мережної підсистеми ОС

3. Підвищення ефективності обробки пакетів даних (збільшення максимальної кількості пакетів, які можуть бути оброблені за одиницю часу, шляхом зменшення відповідних обчислювальних витрат (циклів CPU) та обсягів пам'яті).

Приклади (3):

- розпаралелення обробки пакетів на різних рівнях стеку протоколів (stack parallel processing),
- прогнозування значень полів заголовку пакетів (header prediction),
- зменшення кількості копіювань вмісту внутрішніх буферів,
- розвантаження ядра за рахунок обчислення контрольних сум мережним адаптером (checksum offload).

02.1. Організація роботи мережної підсистеми ОС Linux

1. Робота мережної підсистеми ОС Linux базується на принципі реагування на події різними модулями на противагу єдиному потоку управління роботою всієї підсистеми.

2. Приклади подій:

- системні виклики від користувацьких процесів та інших модулів ядра,
- апаратні переривання від мережного адаптера і таймера,
- програмні переривання, тощо.

02.2. Організація роботи мережної підсистеми ОС Linux

- Мережна підсистема ядра, теоретично, майже уся може виконуватись в режимі користувача: для таких операцій, як формування пакетів TCP/IP ніякі привілеї режима ядра не потрібні.

- Проте у сучасних ОС (тим більше у ОС, які встановлюються на високонавантажених серверах) від мережного стеку вимагається максиальна продуктивність.

02.3. Організація роботи мережної підсистеми ОС Linux

- Мережна підсистема, яка б працювала у режимі користувача, була б змушена постійно звертатись до ядра для «спілкування» з мережним обладнанням, що призвело б до суттєвих накладних витрат і відповідного зниження продуктивності.

- Тому мережні підсистеми (як в Linux, так і в інших ОС) розміщують у складі ядра ОС.

02.4. Організація роботи мережної підсистеми ОС Linux

Потоки управління роботою мережної підсистеми ОС Linux можна поділити на три основні групи:

1. Потоки управління в контексті користувацького процесу.

2. Потоки управління в контексті програмних переривань.

3. Потоки управління в контексті апартних переривань.

02.5. Організація роботи мережної підсистеми ОС Linux

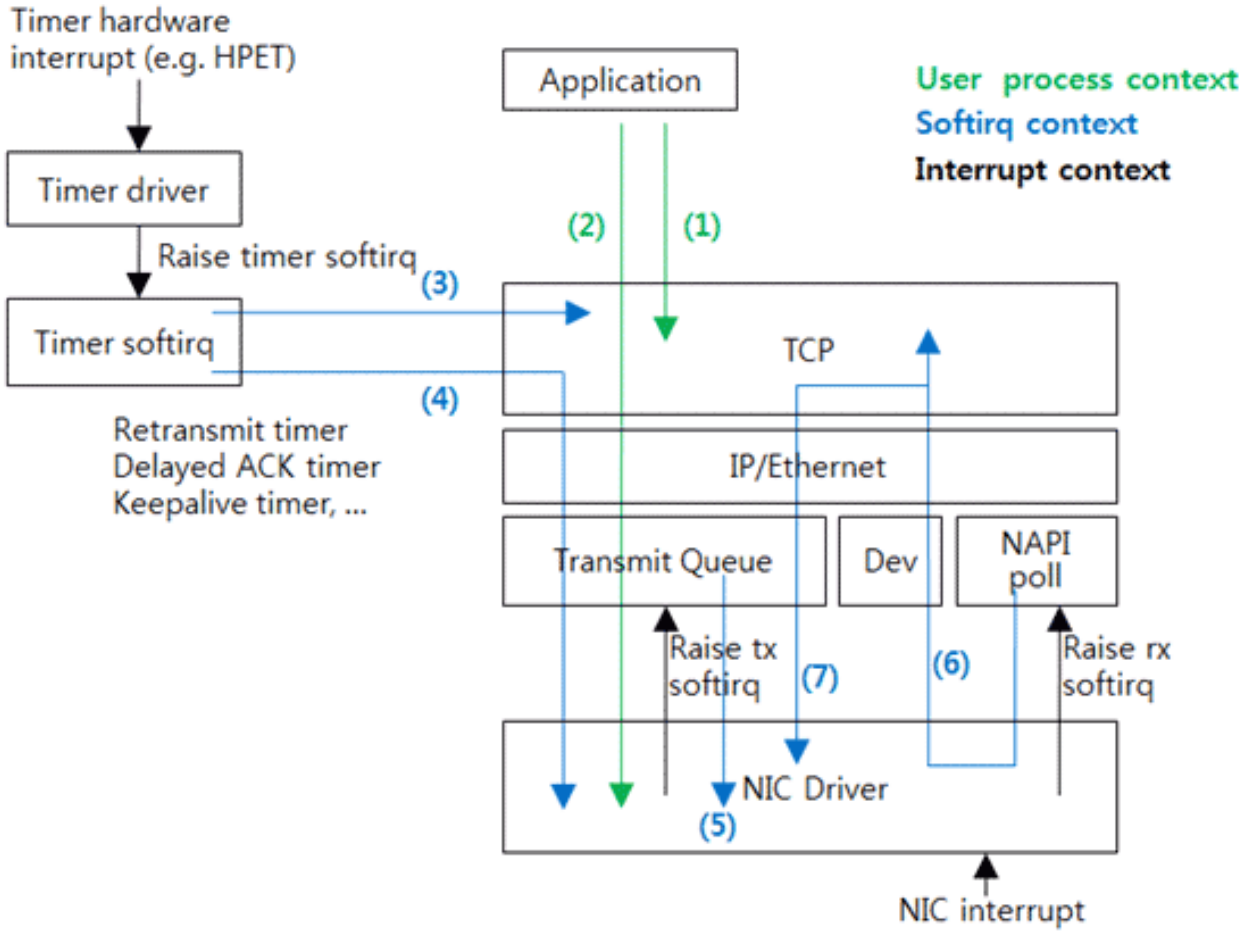


Схема організації роботи мережної підсистеми ОС Linux

02.6. Організація роботи мережної підсистеми ОС Linux

1. Потоки управління в контексті користувацького процесу.

Потік управління (1) займається обробкою отриманого від користувацького процесу системного виклику, який не призводить до передачі пакету даних у мережу (отримання даних, контрольні функції налаштування TCP-з'єднання та ін.).

Потік управління (2) займається обробкою системного виклику, який призводить до передачі пакета даних у мережу.

02.7. Організація роботи мережної підсистеми ОС Linux

(2).1. Дані з користувацького процесу спочатку передаються модулю опрацювання TCP, в якому формуються відповідні пакети даних, які в свою чергу передаються програмним модулям нижчого рівня (IP/Ethernet).

(2).2. Після обробки на цих рівнях пакети даних потрапляють у чергу відправки пакетів (Transmit Queue).

02.8. Організація роботи мережної підсистеми ОС Linux

(2).3. Контролер черги згідно заданого порядку (в термінології Linux: дисципліна черги (qdisc, queue discipline)) скеровує пакети до драйвера (NIC Driver) для перадчі в апаратну пам'ять мережного адаптера (після чого вони надсилаються по мережі отримувачу).

Шляхом зміни дисципліни черги (qdisc) в ОС Linux здійснюється управління мережерним трафіком (по замовченню в якості qdisc використовуються проста схема FIFO).

02.9. Організація роботи мережної підсистеми ОС Linux

2. Потоки управління в контексті програмних переривань.

- Програмні переривання (в термінології Linux: softirq — software interrupt request) — це спеціальний механізм ядра Linux, логіка роботи якого імітує обробку апаратних переривань.
- У випадку мережної підсистеми цей механізм застосовується для виконання її внутрішніх завдань, які не пов'язані напряму із взаємодією з користувацькими процесами.

02.5. Організація роботи мережної підсистеми ОС Linux

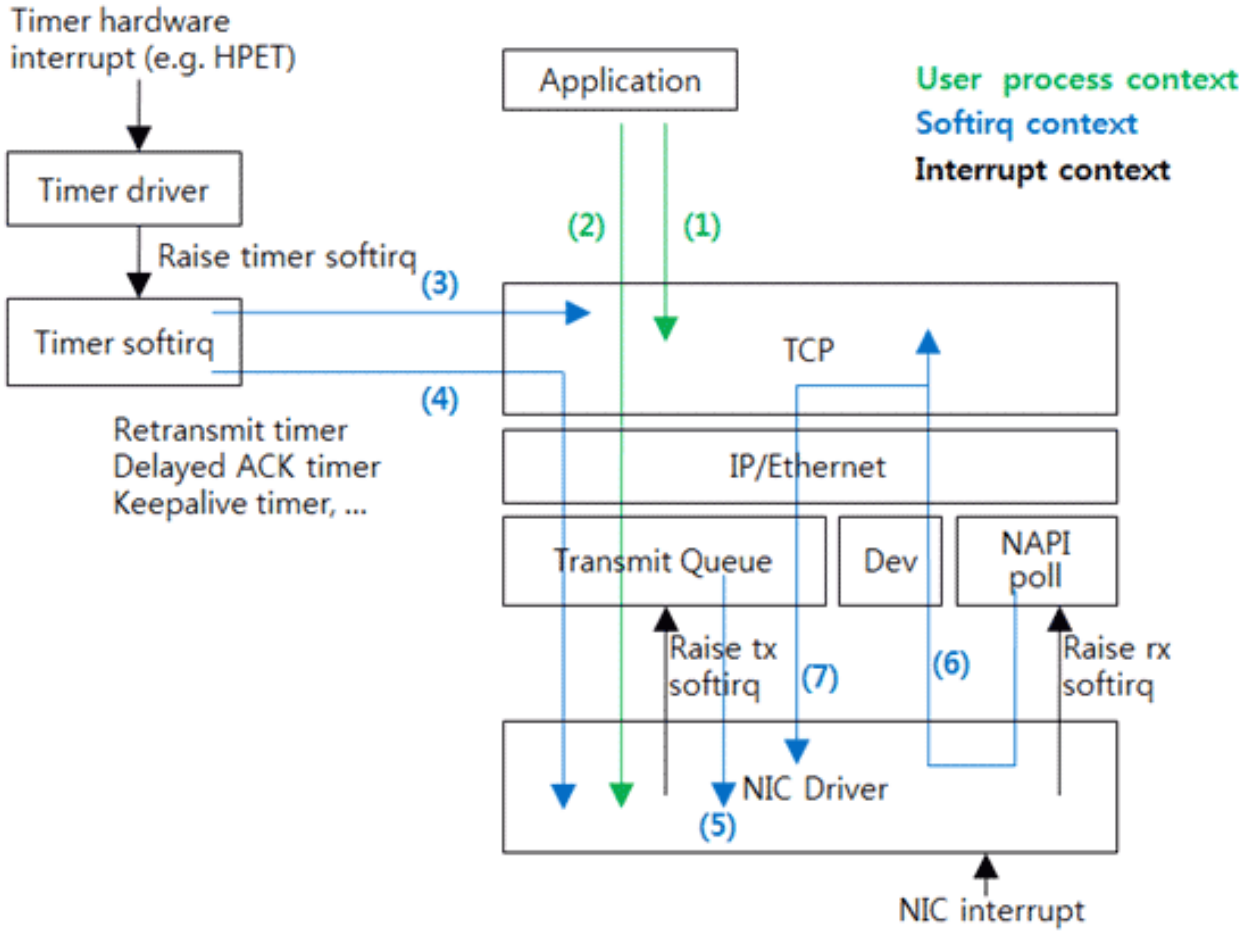


Схема організації роботи мережної підсистеми ОС Linux

02.9. Організація роботи мережної підсистеми ОС Linux

Потік управління (3) займається обробкою спрацювання таймера, встановленого модулем TCP, наприклад, для визначення часу завершення очікування (TIME-WAIT), після якого модуль TCP видалить відповідне з'єдання.

Потік управління (4) так само займається обробкою спрацювань таймера, встановленого модулем TCP, яка, на відміну від (3), призводить до відправки пакета даних (наприклад, повторна відправка пакета після закінчення часу очікування підтвердження (ACK) від отримувача пакета).

02.10. Організація роботи мережної підсистеми ОС Linux

Потік управління (5) займається скеруванням пакетів даних з черги відправки (Transmit Queue) до драйвера (відповідне програмне переривання tx softirq генерується самим драйвером).

Потік управління (6) займається отриманням пакетів даних з використанням механізму опитування (модуль NAPI poll) для розвантаження ядра від обробки великої кількості апаратних переривань по кожному отриманому пакету (механізм опитування (polling) вмикається тоді, коли кількість отримуваних пакетів перевищує задане порогове значення).

Потік управління (7) займається випадками, які потребують додаткової відправки TCP-пакетів.

02.11. Організація роботи мережної підсистеми ОС Linux

3) Потоки управління в контексті апартних переривань.

1. Переривання від таймера (ці переривання відповідно генерують програмні переривання потоків управління (3) та (4)).

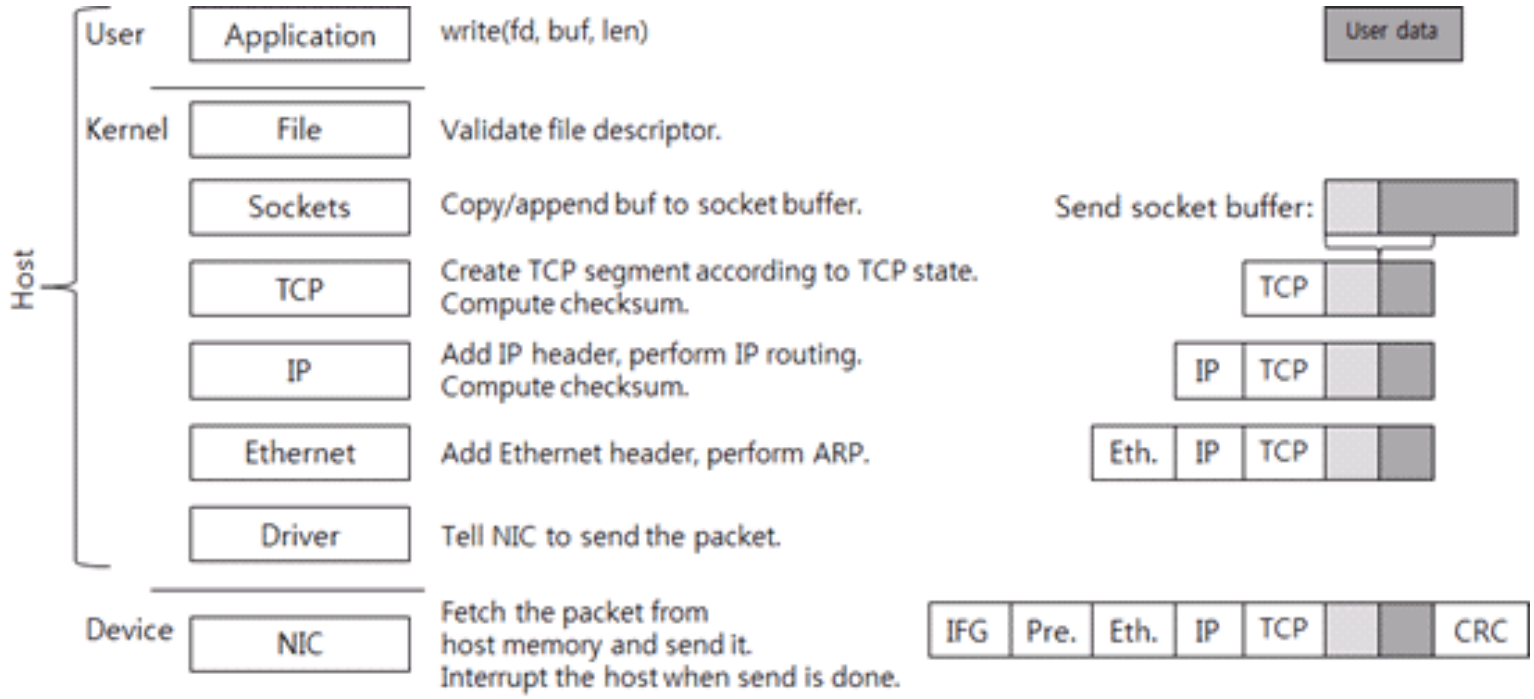
2. Переривання від мережного адаптера (NIC interrupt), які зокрема генерують програмні переривання:
потоку управління (5) (Raise tx softirq — обробка відправки пакетів в мережу) та
потоку управління (6) (Raise rx softirq — обробка отримання пакетів з мережі).

03.1. Відправка та отримання даних мережною підсистемою ОС Linux

1. Відправка даних починається з виклику функції (наприклад, функції write) у користувацькому процесі.

- В якості «пункту призначення» вказується файловий дескриптор (fd) сокета (який був створений раніше з вказанням адреси отримувача). Крім цього вказується які дані треба передати (buf) та їх розмір (len).
- Даний виклик функції запускає обробку відповідного системного виклику в режимі ядра.

03.2. Відправка та отримання даних мережною підсистемою ОС Linux



Основні етапи відправки даних мережною підсистемою ОС Linux

03.3. Відправка та отримання даних мережною підсистемою ОС Linux

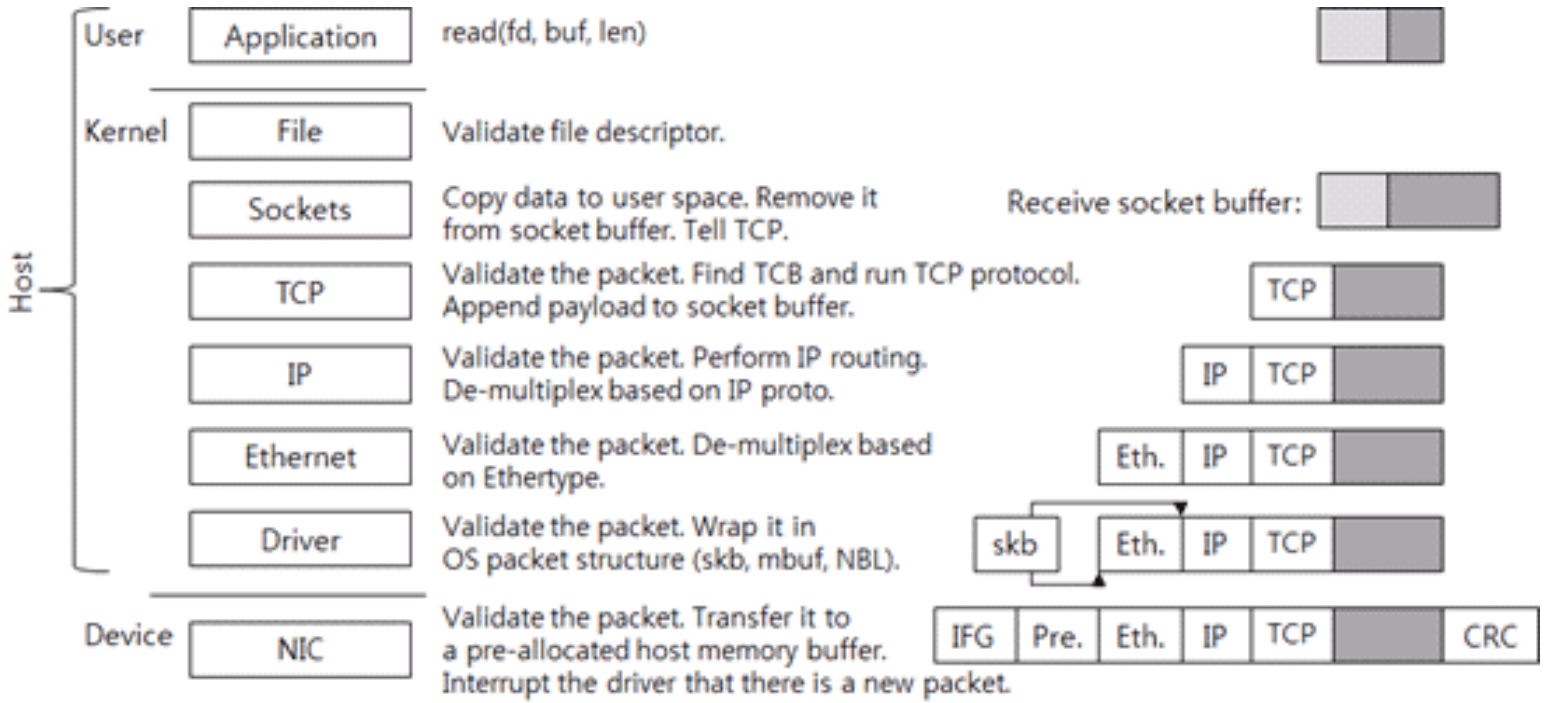
- На рівні роботи з сокетами (Sockets) дані, які треба відправити по мережі, копіюються у буфер для відправки (Send socket buffer).
- Після цього керування передається на рівень транспортного протоколу (наприклад, TCP), далі на рівень IP і на рівень Ethernet. На кожному з цих рівнів відпрацьовує логіка відповідного протоколу та додаються відповідні заголовки до пакета даних.
- На останньому кроці дані передаються драйверу мережного адаптера (Driver), який скеровує їх до мережного адаптера (NIC).

03.4. Відправка та отримання даних мережною підсистемою ОС Linux

2. Отримання даних починаєтья з надходження пакета даних до пам'яті мережного адаптера (NIC), який генерує відповідне апаратне переривання (NIC interrupt). Далі пакет даних скеровується на обробку драйвером (Driver).

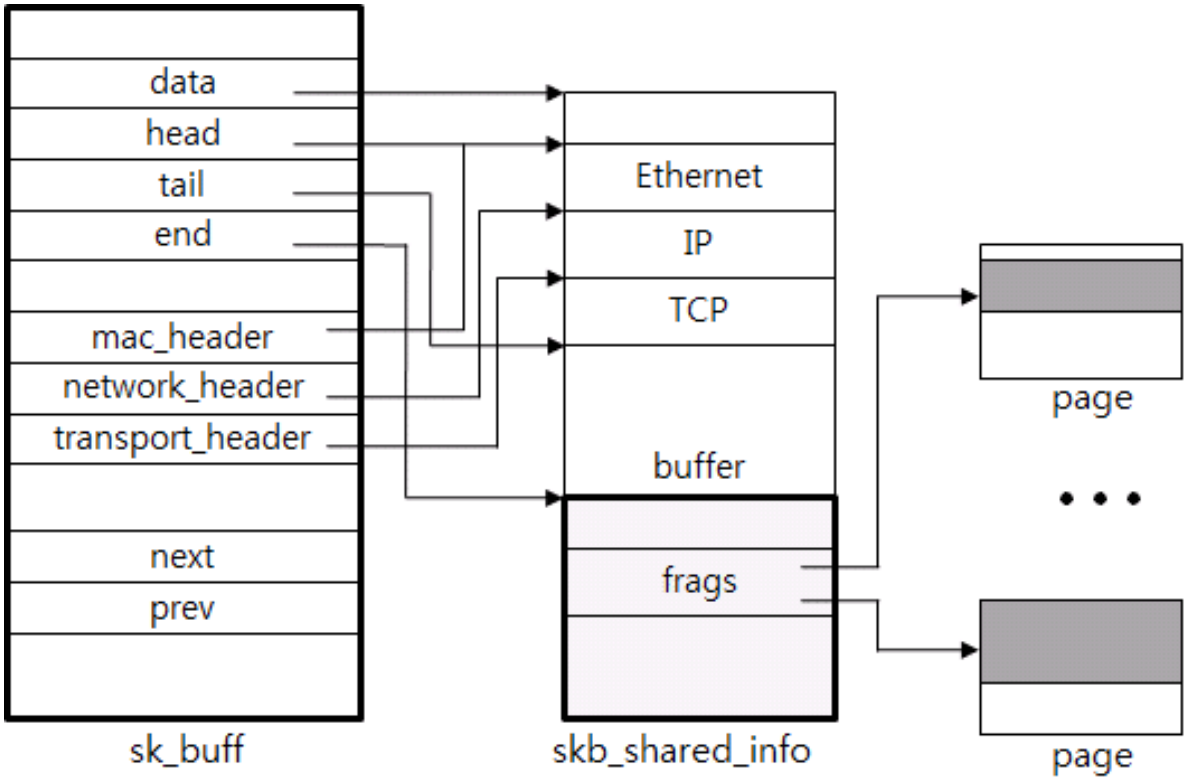
Після цього для отриманого пакета створюється примірник структури sk_buff, яка відображає його структуру (це потрібно для того, щоб наступні вищі рівні обробки знали, що з ним робити).

03.5. Відправка та отримання даних мережною підсистемою ОС Linux



Основні етапи отримання даних мережною підсистемою ОС Linux

03.6. Відправка та отримання даних мережною підсистемою ОС Linux



Структура sk_buff

03.7. Відправка та отримання даних мережною підсистемою ОС Linux

- Пакет передається послідовно рівням обробки: Ethernet, IP, TCP, кожний з яких виконує необхідні дії по обробці пакета згідно логіки роботи відповідного протокола.
- Рівень транспортного протоколу (TCP) розміщує отримані дані у буфер для отриманих даних (Receive socket buffer) рівня роботи с сокетами.
- Рівень роботи з сокетами (Sockets), отримавши від користувацького процесу запит на отримання даних (наприклад у вигляді функції read), копіює дані з буфера для отриманих даних (Receive socket buffer) у структуру даних користувацького процесу (buf).

03.8. Відправка та отримання даних мережною підсистемою ОС Linux

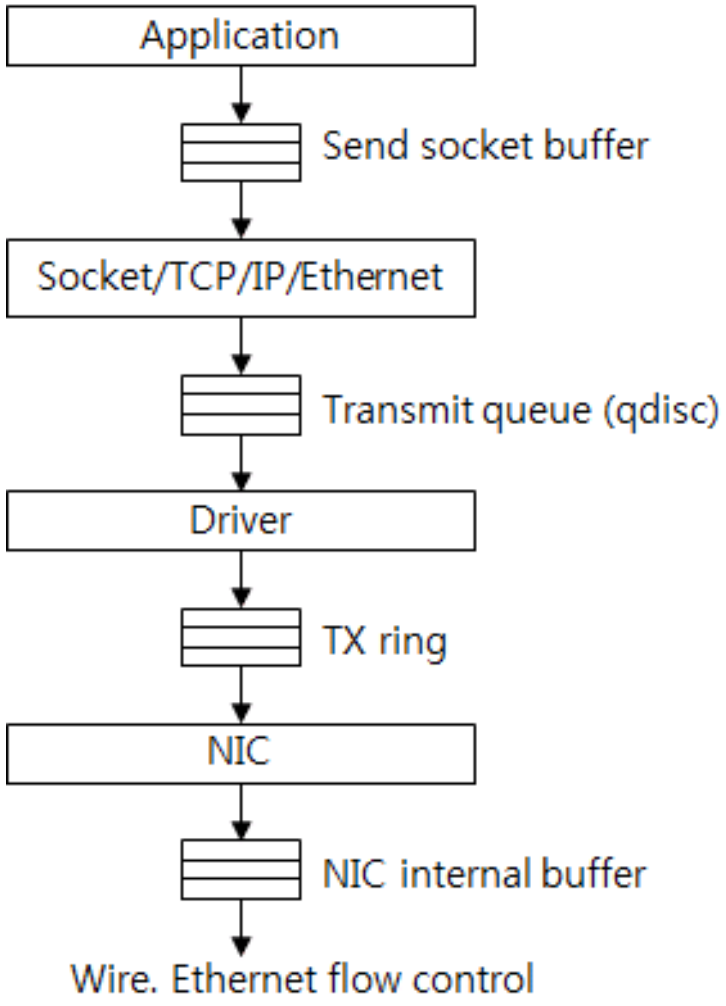
Структури даних ядра, які використовуються для проміжного збереження даних, що відправляються мережною підсистемою:

1) буфер для відправки (Send socket buffer) рівня обробки сокетів,

2) черга відправки пакетів (Transmit Queue),

3) кільцевий буфер драйвера для відправки пакетів (TX ring).

03.9. Відправка та отримання даних мережною підсистемою ОС Linux



Структури даних для відправки пакетів в мережу

03.10. Відправка та отримання даних мережною підсистемою ОС Linux

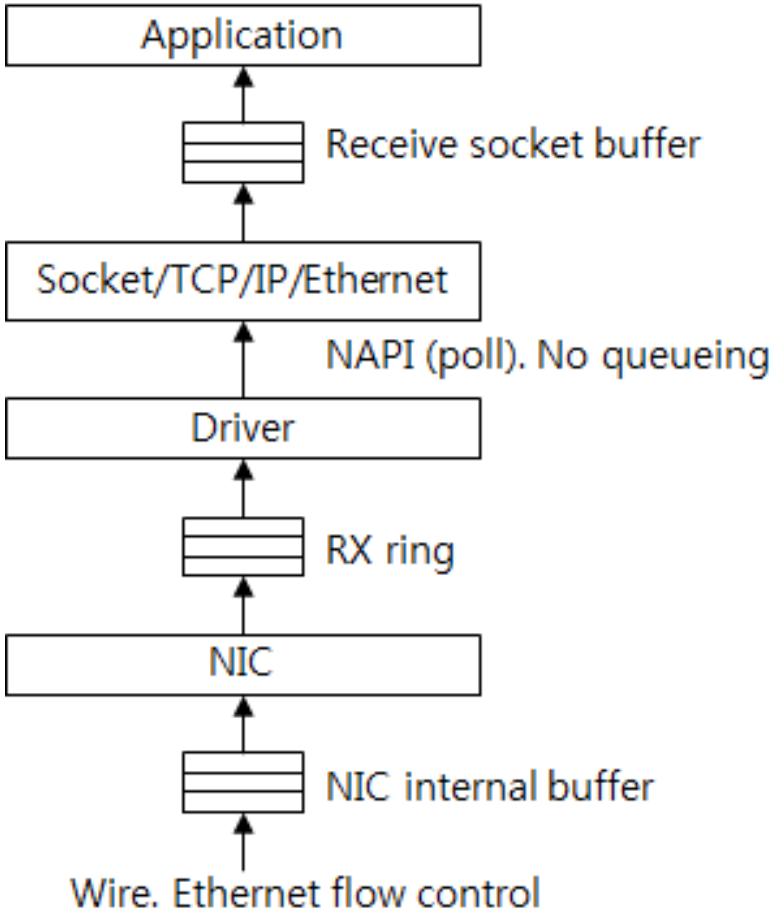
Структури даних ядра, які використовуютсья для проміжного збереження даних, які отримуються мережною підсистемою:

1) кільцевий буфер драйвера для отримання пакетів (RX ring),

2) буфер для отримання (Receive socket buffer) рівня обробки сокетів.

!! Контроль за коректним використанням цих структур даних є одною з важливих функцій мережної підсистеми. !!

03.11. Відправка та отримання даних мережною підсистемою ОС Linux



Структури даних для отримання пакетів з мережі

04.1. Диспетчеризація пакетів даних в мережній підсистемі ОС Linux

- На вузлах мережі з комутацією пакетів, як правило завжди виконується, так званий, диспетчер пакетів даних (network scheduler, packet scheduler).
- Диспетчер пакетів даних визначає послідовність пакетів в черзі відправки (transmit queue) та черзі отримання (receive queue) (тобто визначає в якій послідовності відсилати пакети у мережу та «приймати» з мережі, тобто передавати прийняті пакети на вищі рівні обробки).
- Існує декілька різних програмних реалізацій такого диспетчера для ядер різних ОС, які основані на різних алгоритмах обробки черги пакетів даних.

04.2. Диспетчеризація пакетів даних в мережній підсистемі ОС Linux

- В мережній підсистемі ОС Linux в якості базового диспетчера пакетів даних (=дисципліни черги (qdisc, queue discipline)) використовується дисципліна pfifo_fast, яка реалізує простий принцип FIFO.
- Дана дисципліна використовує три черги з різними пріоритетами: FIFO 0, FIFO 1, FIFO 2.
- Пакети в черзі FIFO 0 мають найвищий пріоритет (тобто будуть відправлені чи отримані першими), пакети в черзі FIFO 2 мають найменший пріоритет.

04.3. Диспетчеризація пакетів даних в мережній підсистемі ОС Linux

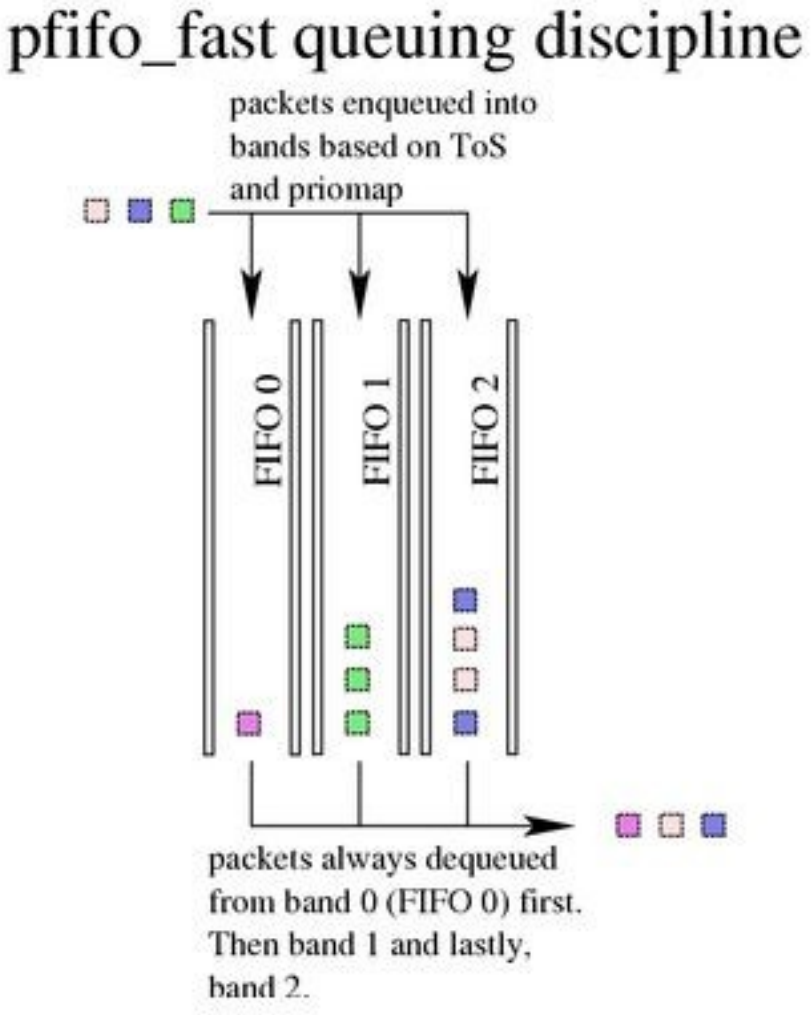


Схема роботи дисципліни pfifo_fast

04.4. Диспетчеризація пакетів даних в мережній підсистемі ОС Linux

- Пакети розкладаються по цим чергам в залежності від значення флагу ToS (Type of Service) в кожному пакеті (значення ToS призначаються пакетам на етапі їх формування на рівні IP).
- Для кожного мережного інтерфейсу можна призначити іншу відмінну від базової (pfifo_fast) дисципліну (схему диспетчеризації пакетів).
- За допомогою цього механізму в ОС Linux виконуєтья управління мережним трафіком (наприклад, обмеження швидкості вихідного трафіку за допомогою дисципліни TBF (Token Bucket Filter) та ін.).

05.1. Мережні інтерфейси в ОС Linux

- Основним об'єктом мережної підсистеми ОС Linux є мережний інтерфейс (network interface).

- Мережний інтерфейс в ОС Linux – це абстрактний іменований об'єкт, що використовується для передачі даних через деяку лінію зв'язку без прив'язки до її реалізації (тобто до низькорівневих деталей її функціонування).

05.2. Мережні інтерфейси в ОС Linux

- Наприклад, якщо в системі є інтерфейс eth0, то у більшості випадків на сучасних комп'ютерах він поставлений у відповідність Ethernet-адаптеру, який вбудований в материнську плату.

- Інтерфейс з іменем ppp0 відповідає за з'єднання «точка-точка» з іншим комп'ютером. Інтерфейс з іменем lo є віртуальним і моделює замкнений сам на себе (вихід підключений безпосередньо до входу) мережний адаптер.

05.3. Мережні інтерфейси в ОС Linux

- Основне завдання мережного інтерфейсу — асбстрагуватись від фізичної складової каналу.
- В такий спосіб програми і ОС будуть використовувати один і той самий метод «відправити пакет» для відправки даних через будь-який інтерфейс (lo, ethX, pppY, тощо), і так само використовувати один і той самий метод «отримати пакет».
- Тобто забезпечується уніфікований API обміну даними, що не залежить від носія.

05.4. Мережні інтерфейси в ОС Linux

- Для визначення того, які мережні інтерфейси присутні в системі, можна скористатись командою: ifconfig -a.

- В наведеному прикладі в системі два активних мережних інтерфейса (eth0 і lo), для кожного з яких виведена інформація про їх стан, налаштування та параметри (а також стан відповідних апаратних засобів для eth0).

05.5. Мережні інтерфейси в ОС Linux

```
$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:11:2F:A8:DE:A4
          inet addr:172.23.2.114  Bcast:172.23.2.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:11 Base address:0x4000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:60 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4707 (4.5 KiB)  TX bytes:4707 (4.5 KiB)
```

Приклад виклику команди ifconfig

05.6. Мережні інтерфейси в ОС Linux

Зокрема,

- поле Link encap містить тип інтерфейсу,
- HWAddr – апаратну адресу пристрою (наприклад, MAC-адресу для Ethernet),
- MTU – максимальный розмір пакетів даних, що відправляються.

Також в окремих полях міститься статистика того, який об'єм даних був відправлений (TX bytes) та отриманий (RX bytes) через відповідний мережний інтерфейс.

05.7. Мережні інтерфейси в ОС Linux

Найбільш часто зустрічаються наступні мережні інтерфейси:

1) eth (Ethernet) – зазвичай відповідає окремому мережному адаптеру;
2) ppp (Point-To-Point) – з'єднання «точка-точка» з іншим комп'ютером;
3) wl або wlan (Wireless) – безпровідний інтерфейс;
4) lo (Loopback) – віртуальний «замкнений» мережний інтерфейс (використовується для взаємодії «по мережі» між користувацькими процесами одного комп'ютера).

05.8. Мережні інтерфейси в ОС Linux

- Командою ifconfig також можна скористатись для зупинки (відключення) та активації (включення) мережного інтерфейсу, а також для зміни його параметрів.

- Про біжучий стан мережних інтерфейсів можна дізнатись командою: netstat -i.

- Також за допомогою команди netstat можна дізнатись про всі наявні мережні з'єднання (у вигляді списку всіх локальних мережних портів транспортного рівня): netstat -a.

05.9. Мережні інтерфейси в ОС Linux

- Відповідно список «активних» мережних з'єднань, по яким в даний момент часу відбувається обмін інформацією, можна отримати командою: netstat -a | grep ESTABLISHED
- Список біжучих мережних з'єднань у вигляді списку IP-сокетів можна отримати командою: lsof -i.
- Також список усіх сокетів можна отримати командою: ss (socket statistics).

05.10. Мережні інтерфейси в ОС Linux

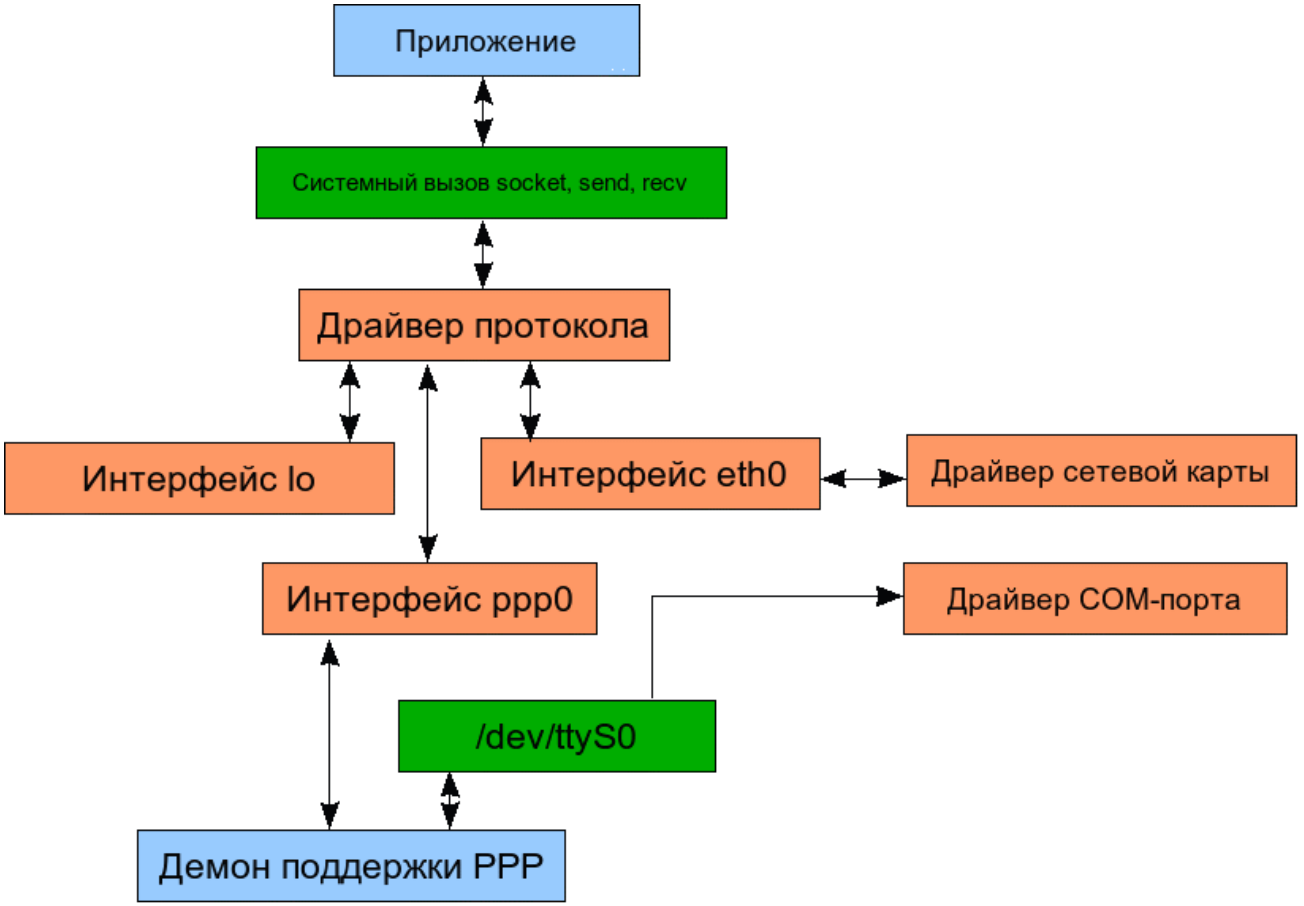


Схема взаємодії модулів та драйверів мережної підсистеми

05.11. Мережні інтерфейси в ОС Linux

Порядок передачі даних в мережу в термінах абстракції мережних інтерфейсів:

1.1) Нехай користувацький процес намагається відправити якісь дані через мережу.

1.2) Перед відправкою даних, за допомогою виклику функцій програмного інтерфесу сокетів Берклі (bind, connect та ін.) буде встановлено потрібне мережне з'єднання. Це з'єднання буде доступне для користувацього процесу у вигляді файлового дескриптора відповідного сокета.

1.3) Тепер кожний байт даних записаний в цей файловий дескриптор буде відправлятись по мережі отримувачу.

05.12. Мережні інтерфейси в ОС Linux

Що відбувається з цими даними в мережній підсистемі?

2.1) Спочатку з даними має справу драйвер мережного протоколу, який:

2.1.1) формує з даних пакети,

2.1.2) визначає через який мережний інтерфейс ці пакети мають відправлятись,

05.13. Мережні інтерфейси в ОС Linux

2.1.3) дописує до пакетів необхідні заголовки,

2.1.4) після чого передає пакети на обробку відповідному мережному інтерфейсу (точніше, ставить пакет в чергу відправки, пов'язану з цим інтерфейсом).

2.2) Далі, в залежності від типу мережного інтерфейсу, над пакетами можуть виконуватись різні дії.

05.14. Мережні інтерфейси в ОС Linux

2.2.1) У випадку інтерфейса lo (loopback) пакет даних забирається з черги на відправку і відразу розміщується в черзі отриманих пакетів, звідки він потрапить до драйверу протоколу.

2.2.2) У випадку інтерфейса eth0 до пакета буде дописано заголовок Ethernet і він буде переданий драйверу мережного адаптера, який проінструктує мережний адаптер, звідки взяти і як відправити цей пакет.

05.15. Мережні інтерфейси в ОС Linux

2.2.3) Ще один варіант – це обробка пакету мережним інтерфейсом PPP (Point-To-Point). Пакет розміщується в черзі відправки інтерфейса ppp0, звідки його забере демон pppd. Демон допише до пакету необхідні заголовки і через спеціальний символьний файл-пристрою dev/ttyS0 передасть пакет драйверу COM-порта, який забезпечить фізичну відправку пакета по послідовному порту.

!! Відповідно, при отриманні даних пакети проходять розглянутий шлях у зворотньому порядку. !!