1 KORIŠTENJE NAPRAVA PREKO OPERACIJSKOG SUSTAVA

1. **Kako se iz programa koriste naprave? Zašto baš tako?**

--Koriste se kroz usluge OS-a jer on nudi sučelje za njihovo korištenje te ih inicijalizira I zna kako preko upravljackih programa njih koristit

2. Kojim sučeljima se mogu koristiti naprave? Koja su osnovna, a koja dodatna?

-- **Osnovnim:** open, close, read, write

--**Dodatnim:** Iseek, select, poll, fsynx,mkond,fcntl,readv….

3. Što su to asinkrone operacije s napravama?

-- asinkrone operacije ne daju povratan odgovor nakon što završe sa radom, tj ne treba ju čekati da završi sa radom

4. Navedite nekoliko mogu´cih podjela naprava ovisno o svojstvima, na˘cinu spajanja, na˘cinu komunikacije, smjeru podataka, ...

Svojstvima: fizička dim, način spajanja na računalo (**unutarnje I vanjsko** (sabirnicama, kablovim)), dohvat podataka (sector, pkaet), Ulazno I ulazno-izlazne (miš, printer))

2 JEDNOSTAVNI MODEL OS-A ZA UPRAVALJNJE NAPRAVAMA

1. **Ako zanemarimo načcin spajanja naprave, koji su osnovni načini upravljanja napravama (posluživanja naprava)?**
   - Radno cekanje, prekidi, izravan pristup memoriji (DMA)

2. **Opisati osnovna nacela upravljanja radnim cekanjem, prekidima te korištenjem sklopova s izravnim pristupom spremniku (DMA).**
   --radnim cekanjem – petlja u kojoj se ceka da naprava bude spremna
   -- prekidi – naprava javlja kad je spremna
   - DMA – naprava sama prenosi podatje u radni spremnik

3. **U jednostavnom modelu jezgre ulazno-izlazne operacije može se pretpostaviti da naprava obavlja zadane joj naredbe slijedno. Stoga bi procesi koji su tražili takve operacije mogli biti u jedno uredenom redu, ¯ čekajući dovršetke svojih operacija. Koji su problemi ovog modela zbog specifičnosti stvarnih sustava? Kako se u stvarnim sustavima rješavaju takvi problemi?**
   -- – naprave možda ne moraju posluživati zahtjeve po redu prispije´ca
   – operacija tražena od procesa može zahtijevati više UI operacija, možda i s više naprava

4. Kako se rješava problem složenosti ostvarenja operacijskog sustava u stvarnim sustavima, npr. Linuxu?
--podjelom na podsustave I slojeve (ona velika mreza linuxa)

3 MODEL NAPRAVA

1. **Kako se može pristupiti napravama, tj. njihovim upravljačkim sklopovima?**
--napravama mozemo pristupiti izravno ili preko drugog medjusklopa/kontrolera. Npr preko USP medjusklopa itd

2. **Zašto su stvarne arhitekture računala hijerarhijski građene, s premosnicima između različitih dijelova (sabirnica)?**
--zato što sporije naprave ako se nalaze na istoj sabirnici kao i brze, mogu utjecati na rad brzih naprava tj usporavati ih pa se grade premosnice tako da su sporije naprave ispod, a brze iznad. Sporije naprave idu na sporije sabirnice.

2. **Nekim se napravama pristupa korištenjem adresa. Koji sve problemi zbog toga mogu nastati, tj. što treba "reći procesoru" u tom slucaju?**

--Ako OS koristi straničenje tada treba tu adresu mapirati u tablicu prevođenja—inace problemi?. Adresu treba upisati i u napravu, a ne samo u prirucni spremnik procesora??

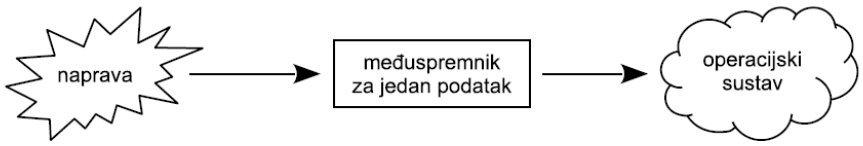
4. **Zašto se koriste međuspremnici? Koju funkcionalnost obavljaju?**

-- • Pojam međuspremnik se obično se odnosi na spremnik koji se koristi za prijenos podataka između dviju strana, npr. naprave i operacijskog sustava.

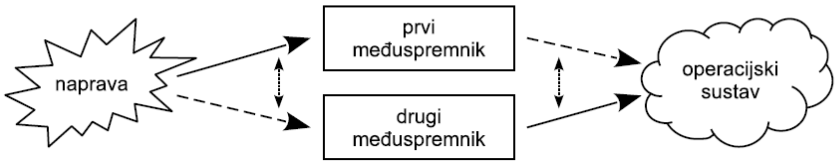
--• Pri prijenosu se podaci micu s one strane koja ih je kopirala u međuspremnik

5. Navesti vrste međuspremnika i opisati kako se oni koriste.

1. Međuspremnik za jedan podatak – u njega stane samo jedan podatak (bajt, paket, blok)
   a) ulazna naprava upiše podatak u međuspremnik
   b) tek nakon toga OS može čitati taj podatak
   c) idući podatak naprava može upisivati tek nakon što OS pročita prethodni
   d) za izlaznu napravu je obrnuto: OS prvi upisuje, naprava čita nakon toga

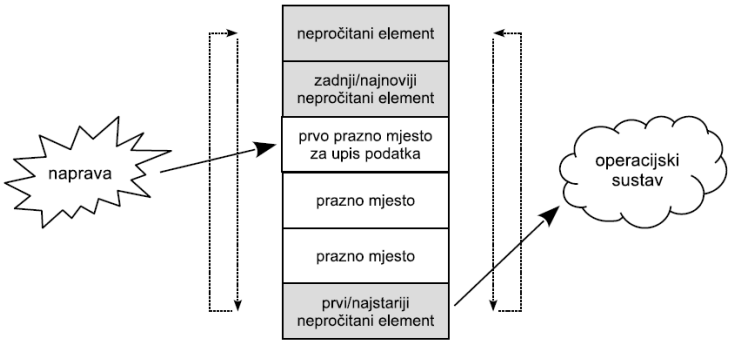


2. Dvostruki međuspremnik (engl. *double buffer*) – u prvi se piše dok se iz drugog čita, uz zamjenu uloga kad se čitanje/pisanje obavi; npr.:
   a) ulazna naprava piše u prvi, a OS može čitati iz drugog međuspremnika
   b) kad su obje operacije gotove, međuspremnici se zamjenjuju: naprava piše u drugi, a OS čita iz prvog
   c) ne moraju operacije biti paralelne, ali onda netko mora čekati; npr. ako je OS pročitao drugi, ali naprava još nije stavila novi podatak u prvi, OS mora čekati; i obratno, kada naprava mora čekati da OS pročita podatak



Slika 3.4. Dvostruki međuspremnik pri komunikaciji s ulaznom napravom

3. Kružno korištenje međuspremnika
   - proširenje koncepta dvostrukog međuspremnika na N međuspremnika
   - kad se jedno mjesto u međuspremniku popuni, idući podatak stavlja se na iduće (sudjedno) mjesto, ako je "prazno"; ako nije mora se čekati
   - kad se dođe do zadnjeg mjesta, iduće mjesto je prvo u međuspremniku (mjesta međuspremnika se kružno obilaze)



6. **Usporediti meduspremnik (engl. *buffer*) i prirucni spremnik (engl. *cache*).**
--međuspremnik se koristi za prijenos podaraka, a cache za ubrzanje rada. U medjuspremniku se spremaju stvarni podaci koji se brisu nakon preuzimanje jedne strane, dok se u cache sprema kopija podataka. Bolje je od brzih naprava koristiti medjuspremnik jer kopiranje podataka moze usporiti operacije koje trebaju biti brze.

4 PRIMJERI NAPRAVA

1. **Navesti osnovna svojstva komunikacije preko serijske veze (npr. RS232, USB,PCIe)**
   --prenosi se bit po bit
   --potrebno je sinkronizirati početak i kraj
   --koriste se 2 vodiča (za svaki smjer po jedan)
   -- jedinica podataka je **znak – grupa uzastopnih bitova**
   -- pri slanju podtaka potrebno je posebnim znakom označiti početak i kraj

2. **Navesti osnovna svojstva komunikacije preko paralelne veze (PCO,ČPT).**

   --bitovi se salju paralelno preko vise linja

   --u jednom ciklusu signala se prenosi vise bitova

   --sikronizacija je potrebna dodatnim signalim linijama nakon svakog bita tj jednog znaka koji je poslan paralelno

   --potrebno je puno linija za sikronizaciju –usporedva rad na vecim udaljenostima →zato se korsiti serijska veza

3. **Navesti nekoliko protokola koji koriste serijsku i nekoliko koji koriste paralelnu vezu.**

- Serijska **: RS232, USB, PCIe, I2C**
- Paralelna: **PCI, LPT**

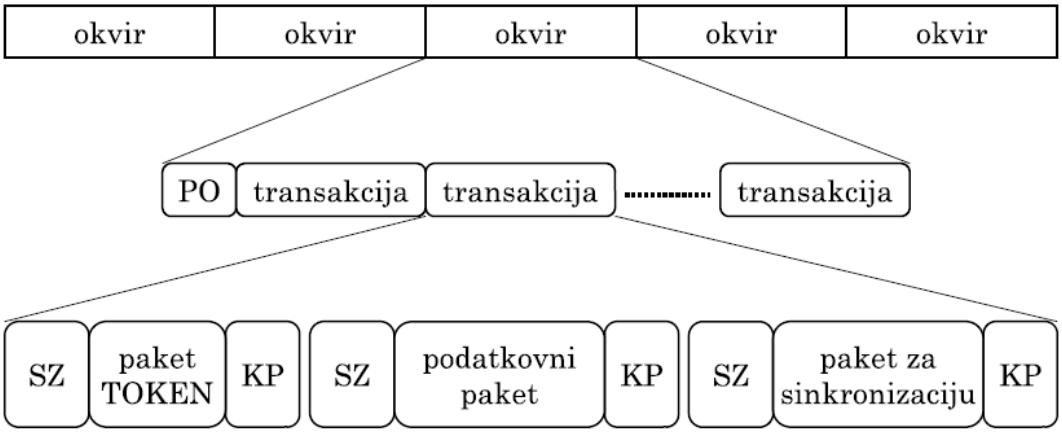
4. **Usporediti serijsku i paralelnu komunikaciju. Koje su prednosti serijske?**
   --- serijska veza je brza od paralelne jer ne treba dodatne sinkronizacije vec samo na pocetku i na kraju te ima manju vjerojatnost gubitka bitova i kasnjenja
5. **Opisati kako su USB naprave spojene u računalu: logički, fizički.**
   --**logicki**: Sve su naprave na jednoj zajednickoj sabirnici
   -- fizicki: naprave su neizravno spojene preko mosta
6. **Opisati osnovni način rada (komunikacije) naprave koja je spojena na USB priključak.**
   • Svaka naprava dobiva (nakon spajanja) adresu: 7-bitovni broj
   • Na jedan upravljac ukupno može biti spojeno 127 naprava
   • Upravljač svaku napravu periodicki proziva – pita ju ima li nešto za javiti, podatke ili status
   • Koristi se period od 1 ms ili kraci, 125 _s za USB 2+

7. Što su to okviri, transakcije, paketi u kontektstu protokola USB?

Okviri: služe za komunikaciju→svaki sadrži oznaku početka okvira i jedna ili više **tranaskcija** koja se sastoji od nekoliko paketa (primjer jedne transakcije koja ima tri paleta : TOKEN (IN, OUT, SETUP,PING), podatkovni (DATA0/1/2) i paket sinkronizacije(ACK,handshake)

Paketi: svaki paket se sastoji od 3 dijela:

1.) Sinkronizacijskog zaglavlja (SZ,sync)
2.) Tjela paketa (paket podatkovne razine)
3.) Oznake kraja paketa (KP, end of packet)



Slika 4.5. Primjer prijenosa okvira, transakcija i paketa

8.Što je to adresa naprave, a što adresa funkcije naprave (kod protokola USB)?

-- Adresa naprave: ona se nalazi u TOKEN paketu prilikom prijenosa podataka (wiki)

-- Adresa funckije naprave: ?

9. Za prijenos podataka USB-om preko protoka (engl. *stream*) koriste se cjevovodi (jednosmjerni): izokroni,prekidni i veliki. Opisati njihova svojstva i namjenu.

--izokroni cjevovodi: garantira se propusnost, ali s mogućim gubicima podataka
--**Prekidni cjevovodi:** za brze odgovore na događaje
-- **Velik prijenosi (BulK) :** za prijenos vece kolicine podataka- ne garantiraju propusnost ni kasnjenja – koriste se slobodni ciklusi na sabirnici

10. **Zašto za mnoge USB naprave nije potrebno instalirati upravljacke programe, vec ih operacijski sustav može koristiti s postojećim?**
--jer OS vec ima klase za vecinu naprava I upravljacke programme za njih (audio naprava, naprave za umrezavanje, pisac, podatkovna , za korisnika (mis, tipkovnica…)

11. Kako se prenose podaci preko PCIe? Koliko vodica se koristi, je li moguc istovremeni prijenos u oba smjera

- Izravna komunikacija
- Serijska komunikacija
- Moze istoremeno u oba smijera (dvosmjerna (full duplex))
- 4 zice

12. Kako se prenose podaci u/iz memorije PCIe naprave?
--prenose se po pojedinim stazama
--Svaka veza izmedu dviju naprava povezanih preko PCIe se sastoji od jedne ili više staza
--za svaki smjer imamo 2 vodica—ukupno 4→ podaci se salju u paketima

13. Usporediti PCI i PCIe. Koje su prednosti sabirnice PCIe?
--PCI – paralelna I spora, jednosmjerna losa sinkronizacija I puno zica , PCIe- serijska I brza I dvosmjerna, dobra sinkr, malo zica

14. Koja je jedinica podataka (ne upravljačkih naredbi) koja se prenosi preko SATA protokola? Zašto nije proizvoljna velicina podataka?

---Serial ATA ima jedinicu podataka **blok/sector –** cijeli disk je logicko polje blokova koje posluzuje sam disk – OS ne mora brinuti o tome gdje je glava, sector, staza za citanje itd..

5 PODRŠKA ZA OSTVARENJE NAPRAVA U LINUXU

1.Unutar jezgre kod se može izvoditi u "razlicitim kontekstima". Koji su to i koja ograničenja postavljaju pojedini konteksti?

Može se izvoditi u :

1) **Jezgrinom kontekstu procesa OGR ?**
2) **Kontekstu jezgrine drteve OGR ?**
3) **Bez konteksta dretve-atomarno** (obrada prekida, alarmi..) OGR: obrada koja je zapocela u jezgrinom nacinu preko prekida ne smije u svom kodu imati dodane blokirajuce pozive npr (CekajSemafor I sl)->mora zavrstiti atomarno. Prve dvije metode bolje—omogucuju bolje performance I brzi odziv na prioritetnij ezahtjeve

2.Što treba napraviti/koristiti ako u jezgrinoj funkciji treba pristupiti adresnom prostoru procesa?
--dohvat korisničkih podataka mora se obavljati korištenjem posebnih funkcija (npr. copy_to_user, copy_from_user)

3.Obrada prekida naprave vrlo je bitan dio upravljanja napravom, ali može bitno utjecati i na svojstva sustava. Zbog cega? Koje mogucnosti u Linuxu stoje na raspolaganju za obradu prekida? Koja su njihova svojstva / kada ih koristiti?

--može bitno utjecati na svojstva sustava jer se izvodi u jezgrenom načinu rada I nije povezana s prekinutom dretvom pa se može odgoditi da je ta prekinua dretva izvodila neki važan posao pa bi trebalo što prije obaviti taj perekid I vratititi se u izvođenje dretve.
-- ako je potrebno više vremena za obradu, onda se rijeli na 2 dijela:

1) **Neophodni dio "top half"** – onaj koji je započeo s obradom prekida ***(atomarni dio)***
2) **Dodatni dio-** obavlja se naknadno s dozvoljenim prekidanjem

   Mogučnosti za dodatni dio:

   a) **Red poslova**
   b) **Višedretvena obrada prekida**
   c) **Lagani prekid**
   d) **Zadačić**

--• odgovarajućim sučeljem se unutar obrade prekida (top half) stvori/pripremi posao koji će se na jedan od navedenih načina odraditi kasnije, nakon završetak ove obrade
• idejno: prva dva načina za duže dodatne poslove, druga dva za kratke
• **prva dva načina (workqueue i threaded IRQ) se preporučuju (u novije vrijeme)**
– obavljaju se u kontekstu posebnih jezgrinih dretvi
– mogu biti odgo¯deni, blokirani i sli˘cno (koristiti blokiraju´ce pozive)

• **druga dva načcina (softirq i tasklet) se izvode atomarno**
– u kontekstu neke dretve koja obavlja takve poslove
– takva dretva obra¯duje zahtjeve nekim redom
– jednom zapo˘ceti moraju biti dovršeni
– ne smiju biti blokirani, tj. koristiti pozive koji bi to mogli izazvati

4.Što se smije a što ne koristiti u jezgri? Je li to ovisi o kontekstu u kojem se izvodi kod jezgre? Kako?
- -Ne smiju se koristiti sinronizacijske funckije, pristup adresnom prostoru procesa I realni brojevi

---to ovisi o kontekstu:

1. **U atomarnom kodu** (obradi prekida/top half, alarmima, taksletima itd):
    a. Ne smijes nikakve fje koje mogu blokirati
    b. Nikakve koje traze kontekst dretvi (sem, monitori)
    c. MOŽEŠ funkcije s radnim čekanjem

2. **U Kodu s kontekstom** (u kontekstu procesa, WORKQUEUE, THREAD):
    a. Sve funkcije (interne jezgine) DA? NE?

5.Što je to modul u kontekstu jezgre Linuxa? Cemu služi?

Upr program može biti na 2 načina uključen u jezgru linuxa:

a) **Permanentno (statički)**
b) Kao modul dinamički – učitava se dinamički I miče kada više nije potreban. Uključuje se I isključuje iz jezgre s naredbama:
    a. Uključivanje: insmod ime-datoteke-s-modulom [arguenti]
    b. Isključivanje: rmmod ime-modula

6.Navesti tri osnovne klase naprava u Linuxu

1) **Znakonvne naprave** **–** daju primaju niz bajtova
2) **blokovske naprave** **–** jedinica podataka je blok, naprave koje ostvaruju datotecne sustave
3) **mrežna sučelja** **–** svrha im je ostvarenje komunikacije

7. Koja je zadaca upravljackog programa naprave?

--upravlja stavrnim ili virtualnim napravama. Mogu imati I prihvat prekida ako za njenu napravu vec to nema ugradjeno u sustav

8. Koja su uobicajena sucelja koja znakovna naprava mora ostvariti, a da bi se uklopila u Linux?