

Оперативни системи и рачунарске мреже

Александар Картељ

aleksandar.kartelj@gmail.com

Рачунарска гимназија

Управљање У/И уређајима

1. Треба да буде што **универзалнији**
 - Пошто може да постоји јако пуно различитих уређаја
2. Што **ефикасније** управљање
 - Сетимо се да су оних хиљадама (чак милионима пута) спорији од процесора
3. Треба да буде **сигурно**
 - Обично се одвија у кернелском режиму
 - Помоћу системских позива их даље користи кориснички програм
4. **Хијерархијски** организовано
 - A. Најнижи ниво је уређај тј. хардвер (физички ниво)
 - B. Потом интерфејс за везивање са остатком рачунара (физички ниво)
 - C. Потом драјвери (логички ниво)
 - D. Коначно кориснички програми (логички ниво)

A. Хардверске компоненте

- Подела према начину преноса података:
 1. Блок уређаји – податке обрађују у блоковима фиксне дужине
 - Сви трансфери базирани на пребацивању одговарајућег блока
 - Могу се адресирати, тј. додељује им се адресни опсег
 - Примери: дискови, DVD-ROM, USB, ...
 2. Карактер уређаји – подаци су у форми тока неодређене дужине
 - Немају директан приступ меморији и нису адресибилни
 - Примери: штампачи, мрежни интерфејси, мишеви, тастатуре, ...

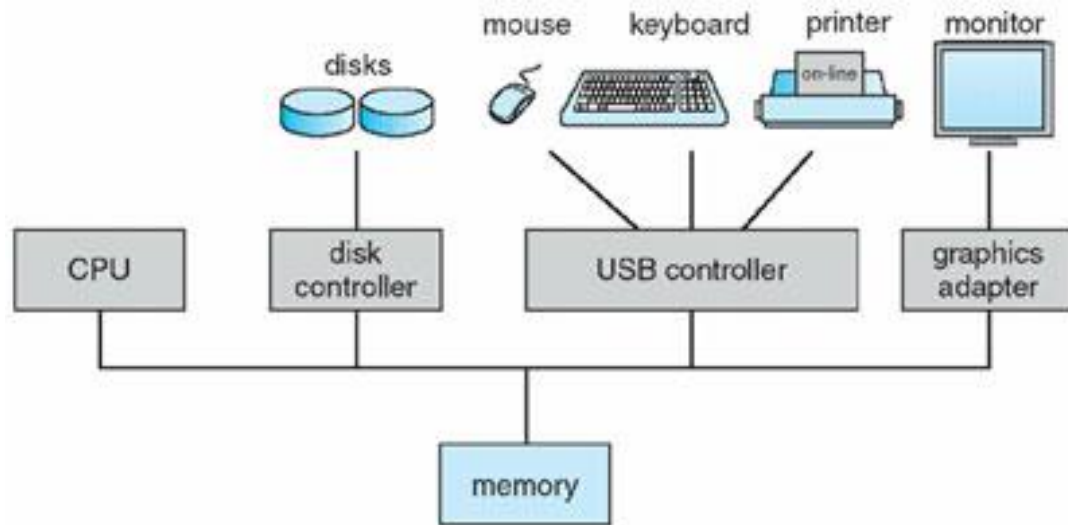
V. Интерфејс уређаја

- Различити начини повезивања уређаја на рачунар
- Спорији имају обично једноставније везивање
- Повезивање је омогућено помоћу **прикључака (конектора)**
 - Не мора путем кабла, може да буде и бежична
 - Комуникација се даље врши путем магистрала
 - Магистрала је скуп комуникационих линија које повезују уређаје
 - Обично постоји више магистрала, а најпознатије су:
 - Системска магистрала – повезује процесор са меморијом и брзим У/И уређајима
 - Спољашње магистрале – повезују спорије уређаје, попут тастатуре, миша, штампача, ...

Контролер уређаја

- Мали процесори са интерном меморијом који контролишу рад уређаја (много једноставнији од CPU)
- Меморија обично сачињена из низа регистара:
 - Регистар примљених података – за податке преузете са уређаја
 - Регистар послатих података – подаци које треба пренети на уређај
 - Статусни регистар – описују тренутно стање уређаја
 - Контролни регистар – овде CPU уписује команду коју уређај треба да изврши
- CPU комуницира са уређајем путем ових регистара
- Други начин је да се ови регистри „угњезде“ у стандардни адресни простор меморије рачунара
 - На тај начин их CPU посматра као део главне меморије

Контролер уређаја (2)



- Нпр. варијанта комуникације са штампачем би могла да буде оваква:
1. CPU шаље садржај за штампу
 2. Контролер уписује тај садржај у регистар послатих података
 3. CPU шаље команду за штампу и ово се уписује у командни регистар
 4. Када је статусни регистар у одговарајућем стању, тј. **спреман**, контролер активира штампање и статус прелази у стање **штампања**
 5. По завршетку, статус се мења поново нпр. у стање **спреман**

Техника прозивања (енг. Pooling)

- Ово је најједноставнији начин управљања У/И уређајима
 - Подразумева да процесор (CPU) уради сав посао
 - Тако што стално проверава статусни регистар уређаја
- На претходном примеру то би значило да:
 - Процесор итеративно проверава да ли је уређај у стању спреман
 - А ако није, позив враћа грешку
 - Када постане спреман, CPU му шаље податак за штампу
 - Поново чека док год уређај не постане спреман да прими податак
 - Потом шаље команду за штампу
 - И чека да се поново појави одговарајући статус
- Проблем са овим приступом је:
 - Што процесор све време проверава уређај, уместо да ради неки други посао...

Прекиди

- Ово је боље од „прозивања“
 - Уређаји хардверски имају могућност да сами сигнализирају промену стања
 - На тај начин процесор не мора стално да их „пита“ у ком су стању
 - Процесор стога има времена да обавља друге активности
- Систем заснован на прекидима функционише овако:
 1. Процес иницира операцију на уређају обраћајући се његовом драјверу
 2. Драјвер преко контролера прослеђује операцију уређају
 3. Када уређај заврши се операцијом, контролор шаље сигнал процесору преко **линије за прекиде (ово је хардверска веза – жица)**
 4. Процесор прекида тренутно извршавање процеса и обрађује прекид
 5. Након обраде прекида, прекинути процес може да се настави

Директан меморијски приступ (енг. DMA)

- Систем прекида ефикасан ако се прекиди не дешавају често
 - Нпр. тастатура би изазивала прекид на сваки притисак тастера
 - У том случају је неефикасно, јер се на сваки притисак тастатуре прекида тренутни процес и обрађује притисак тастатуре
- Директан меморијски приступ (DMA) је хибрид прозивања и прекида
 - Постоји DMA контролер, са својим регистрима
 - Контролер преузима активности чекања на прекиде уређаја од процесора
 - У случају тастатуре, DMA контролер ослушкује тастере активно (прозивањем) док CPU остаје тако слободан за друге операције

С. Драјвери

- Омогућавају униформност улазно-излазног подсистема према кориснику
 - То значи ако апликација треба да одштампа податке, она не мора да зна ком типу штампача се обраћа, нити како он то ради
- Уређаји се групишу у класе и за сваку класу постоји скуп функција
 - Нпр. Класа **штампачи** има стандардизован скуп функција без обзира на модел, генерацију штампача итд.

С. Драјвери (2)

- Могући начини груписања:
 - По смеру преноса:
 - улазни (миш, тастатура)
 - излазни (монитор, штампач)
 - улазно-излазни (диск, мрежна карта)
 - По величини јединице:
 - блок уређаји
 - карактер уређаји
 - По начину употребе:
 - за чување података (диск, флеш меморија)
 - за пренос података (мрежна карта, модем)
 - за интеракцију са корисником (миш, тастатура, монитор)

Планирање улазно-излазних операција

- Временско планирање - редослед извршавања тако да систем буде што ефикаснији
- Може да буде:
 - По редоследу стицања захтева од апликација – неефикасно
 - Обично је критеријум минимизација времена одзива
 - ОС одређује редослед тако да просечно време које прође од слања захтева до добијања услуге буде што мање

Баферовање

- Део меморије у који се привремено смештају подаци пренети од уређаја ка апликацији (или другом уређају)
- Разлози употребе:
 1. Уређаји који сарађују имају различите брзине
 - Нпр. Bluetooth и диск рачунара комуницирају, и диск треба да упише оно што Bluetooth (као спорији уређај) добија бежичним путем
 - Да нема бафера, сваки податак који Bluetooth пошаље би се уписивао на диск
 - То би изазвало да диск стално чека на Bluetooth и не може да ради ништа друго
 - Овако, Bluetooth пуни бафер, и само повремено се они пишу на диск, а бафер празни

D. Кориснички процеси

- На врху хијерархије управљања У/И уређајима су кориснички процеси
- Управљање уређајима се овде омогућава **системским позивима**
- Системски позиви могу бити:
 - Блокирајући
 - Када процес позове блокирајућу функцију, даље извршавање процеса је суспендовано, и наставља се тек по успешном враћању резултата системског позива
 - Нпр. чекање на читавање карактера са улаза (конзоле) – команда **scanf** у С-у, или **Console.Read()** у С#-у.
 - Неблокирајући
 - Процес не чека завршетак системског позива, већ се нпр. враћа грешка одмах
 - Нпр. читавање карактера са улаза, а ако нема карактера, враћа се грешка без чекања