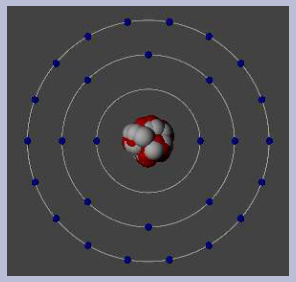
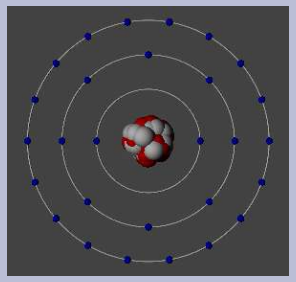
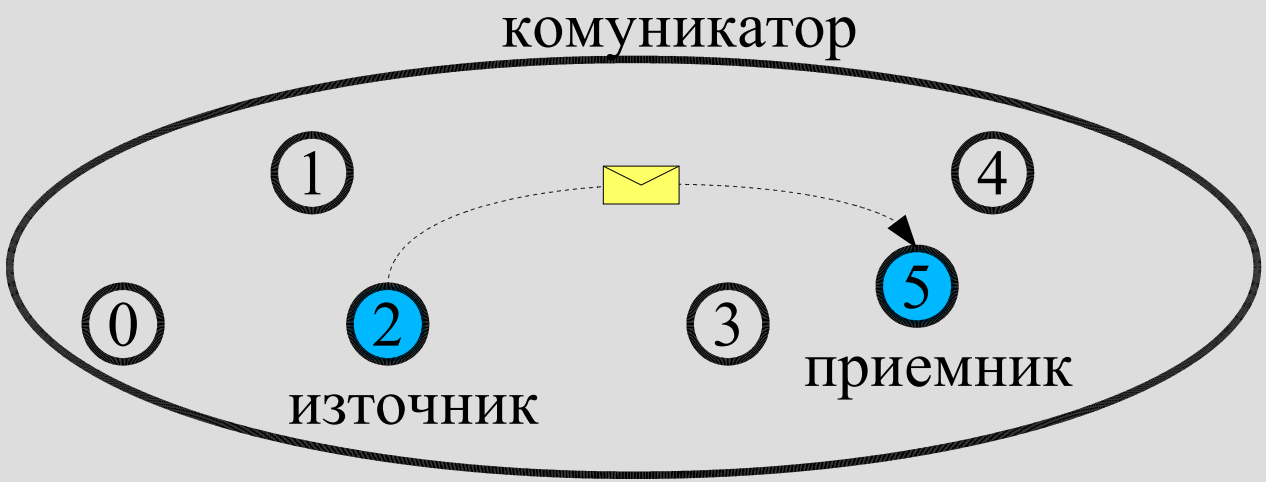


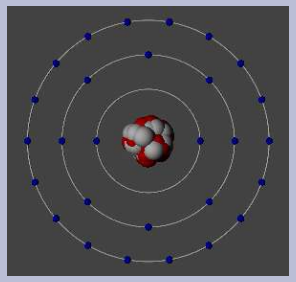
Комуникация от точка до точка



Общи положения

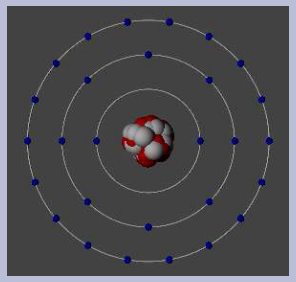
- Комуникациите от точка до точка изискват наличието на точно два процеса:
  - процес източник
  - процес приемник
- Всеки от двата процеса извиква съответните MPI примитиви





Режими на комуникация

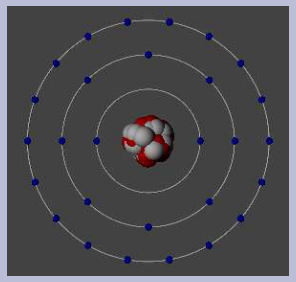
- MPI предлага 4 комуникационни режима:
  - стандартен
  - синхронен
  - буфериран
  - режим с готовност
- Режимите се отнасят до операцията изпращане
- Стандартен, синхронен, буфериран:
  - различават се само по отношението на приключването към приемането на съобщението



Режими на комуникация (2)

	Условие за приключване
Синхронно изпращане	Приключва само след приключване на операцията по приемане
Буферирано изпращане	Винаги приключва (ако не възникне грешка), независимо дали приемането е приключило
Стандартно изпращане	Или е синхронно, или е буферирано (в зависимост от реализацията)
Изпращане с готовност	Винаги приключва (ако не възникне грешка), независимо дали приемането е приключило
Приемане	Приключва след получаване на съобщението

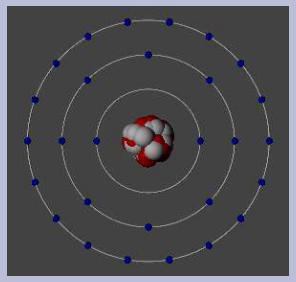
- Режимите на комуникация имат две форми:
  - блокираща
  - неблокираща



Блокиращи комуникации

- Връщането от подпрограмата става след приключване на операцията

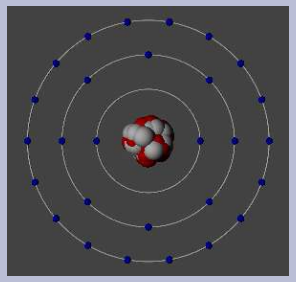
	Блокираща форма
Стандартно изпращане	**MPI_Send**
Синхронно изпращане	**MPI_Ssend**
Буферирано изпращане	**MPI_Bsend**
Изпращане с готовност	**MPI_Rsend**
Приемане	**MPI_Recv**



Стандартно изпращане

MPI_Send (buf, count, datatype, dst, tag, comm)

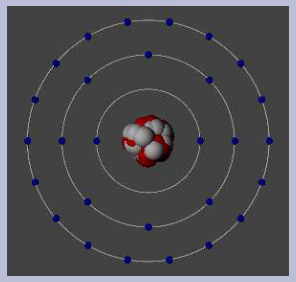
- Приключва след изпращане на съобщението (което може още да е в процес на предаване)
- На практика се реализира се като синхронно или буферирано предаване
- Изпратени, но непрочетени съобщения се натрупват в системата и могат да причинят проблеми
- След приключване на операцията буферът може да се преизползва



Синхронно изпращане

MPI_Ssend (buf, count, datatype, dst, tag, comm)

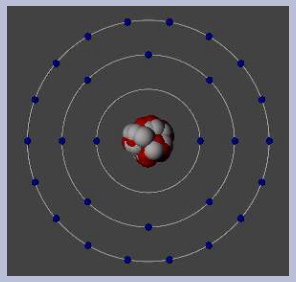
- Приключва след потвърждаване на получаването на съобщението (както писмата с обратна разписка)
- Най-бавната операция за изпращане, но има синхронизационен ефект
- Няма съобщения в транзит между отделните извиквания
- Улеснява откриването и отстраняването на грешки



Буферирано изпращане

MPI_Bsend (buf, count, datatype, dst, tag, comm)

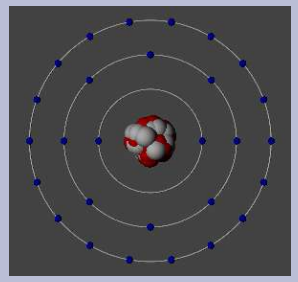
- Приключва веднага, като копира съобщението в буфер за по-късно доставяне
- Необходимо е да се прикачи буфер с MPI_Buffer_attach (buffer, size)
- След употреба буферът трябва да се разкачи с MPI_Buffer_detach (buffer, size)
- Предсказуемо поведение при претоварване на комуникационната система – просто дава грешка



Изпращане с готовност

MPI_Rsend (buf, count, datatype, dst, tag, comm)

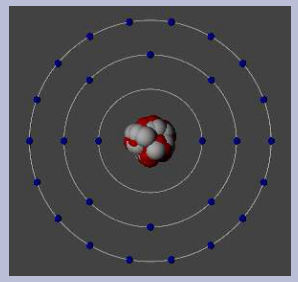
- Приключва веднага след извикване на примитива
- Съобщението ще бъде получено само ако приемащият процес вече е стартирал операция за получаването му, в противен случай поведението е недефинирано (изпускане на пакета, грешка или др.)
- Изисква специфична програмна логика
- Труден за настройка режим
- По-висока производителност



Блокиращо приемане

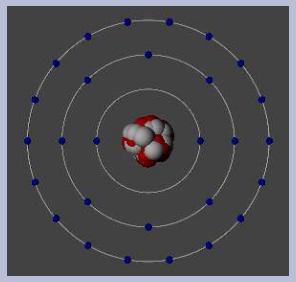

```
MPI_Recv (buf, count, datatype, src, tag, comm, status)
```

- Приключва след получаване на съобщението
- Приемащият буфер трябва да е достатъчно голям (може и по-голям)
- Маски – MPI_ANY_SOURCE и MPI_ANY_TAG
- status съдържа информация за полученото съобщение:
  - източник: status.MPI_SOURCE (C)
    status(MPI_SOURCE) (Fortran)
  - маркер: status.MPI_TAG (C)
    status(MPI_TAG) (Fortran)



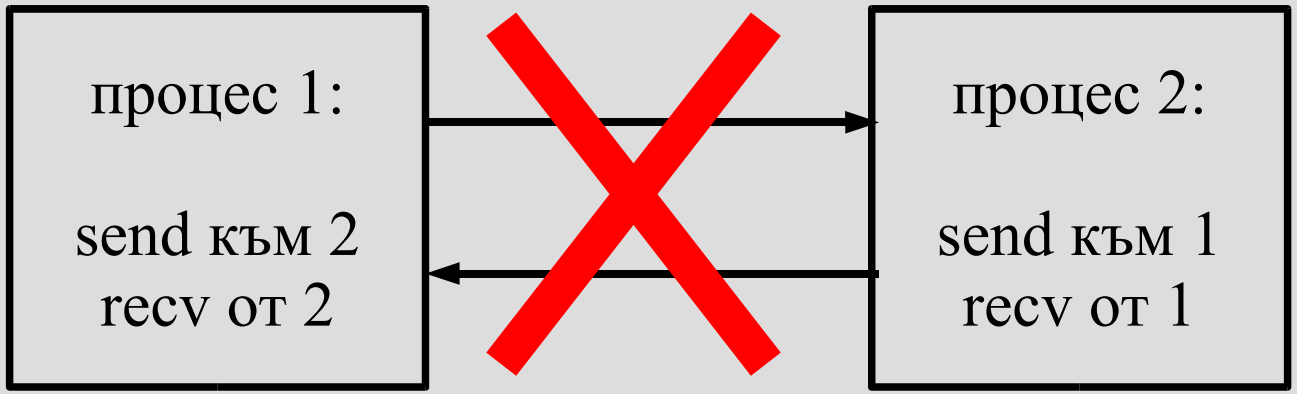
Семантика на комуникациите

- Изпратените съобщения се получават в реда на изпращането им
- Невъзможно е при наличие на съвпадащи операции на изпращане и на получаване и двете да „увиснат“- поне едното ще приключи нормално



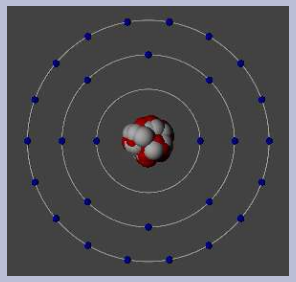
Дизайн на комуникационните схеми (1)

- При използване на стандартно изпращане не трябва да се допуска, че изпращането ще приключи преди да започне приемането



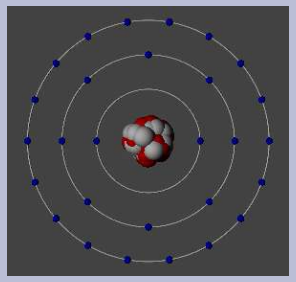
взаимно блокиране

- Проблемът се избягва с използване на MPI_Sendrecv



Дизайн на комуникационните схеми (2)

- При използване на стандартно изпращане не трябва да се допуска, че изпращането ще завърши след като започне приемането.
  - Не трябва да се разчита, че изпратено преди това съобщение е пристигнало някъде, за коректната интерпретация на следващи съобщения.
  - Когато процесите са повече от един е възможен недетериминизъм във времево отношение.



Дизайн на комуникационните схеми (3)

- Всички изпратени съобщения трябва да се „изконсумират“, тъй като в противен случай остават в комуникационната система и могат да я „задръстят“.
- Неспазването на трите правила при дизайна на комуникационните схеми може да доведе до непредсказуемо поведение и/или непреносимост между различните MPI реализации.