

Глава 5. Библиотека PVM сообщений

В пакете Scilab есть возможность работать с PVM параллельной виртуальной машины, что позволяет распараллеливать вычисления, а также выполнять задания удаленно.

PVM (Parallel Virtual Machine) - общедоступная библиотека, позволяющая управлять процессами с помощью механизма передачи сообщений. Существуют реализации PVM для различных платформ.

PVM + SCILAB

Для совместной работы с пакетом Scilab необходимо использовать версию PVM не ниже, чем 3.3.7

Замечание: Пакет PVM должен быть установлен до его использования в Scilab.

Стратегия освоения:

1. осваиваете пакета Scilab (особенно библиотек вычислений, поскольку без них распараллеливание в среде Scilab не имеет никаких выгод в сравнении с классическими языками программирования C и Fortran)
2. осваиваете PVM
3. применяете полученные навыки в Scilab и PVM совместно

В пакет Scilab включены наиболее важные функции библиотеки сообщений PVM. Они основаны на использовании версии 3.3.7 библиотеки PVM.

Замечание:

Целесообразно перед изучением pvm-свойств Scilab освоить прикладную библиотеку Scilab, чтобы имело смысл что-либо распараллеливать таким образом.

К сожалению:

- Не обнаружено никакой документации по совместному использованию PVM и Scilab кроме списка команд и описания их синтаксиса (смотри manual.pdf).
- Нет и хорошей полной документации по работе с PVM.

Все это существенно затрудняет быстрое освоение данного раздела.
Бесплатное Software - есть бесплатное Software...

Команды пакета Scilab (версия 2.6 и 2.7) для совместной работы с PVM

1. **pvm_addhosts** - добавляет хосты в виртуальную машину
2. **pvm_barrier** - блокирует вызов процессов до тех пор, пока все процессы в группе его не вызовут
3. **pvm_bcast** - передает сообщение всем членам группы
4. **pvm_bufinfo** - возвращает информацию о буфере сообщений
5. **pvm_config** - посылает сообщения
6. **vm_delhosts** - уничтожает хосты из виртуальной машины
7. **pvm_error** - печатает сообщение, описывающее ошибку, возвращенную PVM call. group.
8. **pvm_exit** - сообщает локальному pvmd (pvm демону), что данный процесс покидает PVM
9. **pvm_f772sci** - конвертирует комплексное из F77 в комплексный скаляр
10. **pvm_get_timer** - дает системное представление о текущем времени
11. **pvm_getinst** - возвращает копию номера в группе PVM процессов
12. **pvm_gettid** - возвращает идентификатор процесса (tid), отождествленного с именем группы и номером записи
13. **pvm_gsize** - возвращает число членов названной группы
14. **pvm_halt** - остановка PVM демона
15. **pvm_joiningroup** - включение из списков вызывающего процесса для названной группы
16. **pvm_kill** - завершение (прекращение) заданного PVM процесса
17. **pvm_lvgroup** - исключение из списков вызывающего процесса для названной группы
18. **pvm_mytid** - возвращает идентификатор процесса (tid)
19. **pvm_parent** - возвращает идентификатор процесса (tid), вызвавшего данный процесс
20. **pvm_probe** - проверка прибытия сообщения
21. **pvm_recv** - получение сообщения
22. **pvm_reduce** - выполняет сокращение операций над членами определенной группы
23. **pvm_sci2f77** - конвертирует комплексный скаляр в F77
24. **pvm_send** - непосредственно посылает (или многоадресно передает) данные
25. **pvm_set_timer** - устанавливает системные представления о текущем времени
26. **pvm_spawn** - запускает новые Scilab процессы
27. **pvm_spawn_independent** - запускает новые PVM процессы
28. **pvm_start** - запуск PVM демона
29. **pvm_tasks** - возвращает информацию о задачах, запущенных на виртуальной машине
30. **pvm_tidtohost** - возвращает хост заданного PVM процесса
31. **pvmd3** - PVM демон