```
   GNU/Linux [] spm.3.110602.trial.A.python
                                        (Trial Edition)
                               Spm.Python 3.110602 / Python 2.4.6
                                [GCC 4.4.3 (64 bit) on linux2]

                                             NOTE
                           >>>>       Trial period ends at        <<<<
                           >>>> 24:00 hrs (Pacific Standard Time) <<<<
                           >>>>         July 28,      2011        <<<<

                 Type "help", "copyright", "credits", "license" or "spm.Api()" for more informati
                          Type "spm.DemoExtract(dirname = ...)" to extract demo scripts.

                           Please visit www.mbasciences.com for the latest and growing
                        collection of scripts and technical briefs classified in terms of
                                      parallel management patterns.

●  >>> import pool
   >>> import demo
   >>> taskApi     = spm.util.coprocess.shell.policyA;
   >>> taskApiArgs = \
         [
           dict(cmd     = "echo `hostname`",
                timeout = spm.util.timeout.after(seconds = 1)),

           dict(cmd     = "echo `hostname`",
                timeout = spm.util.timeout.after(seconds = 2)),
         ];
   >>> taskTimeout = spm.util.timeout.after(seconds = 10);
✓  >>> demo.main(pool        = pool.intraAll(),
                 taskApi     = taskApi,
                 taskApiArgs = taskApiArgs,
                 taskTimeout = taskTimeout)
     #: MetaStatus (hub): Waiting - ForSpokes [ for up to 30 secs ]
     #: MetaStatus (hub): Tasks   - Eval
           stdOut    : lusaka
           stdOut    : lusaka
     #: MetaStatus (hub): Tasks   - EvalDone
✓  >>> demo.main(pool        = pool.intraOnePerServer(),
                 taskApi     = taskApi,
                 taskApiArgs = taskApiArgs,
                 taskTimeout = taskTimeout)
     #: MetaStatus (hub): Waiting - ForSpokes [ for up to 30 secs ]
     #: MetaStatus (hub): Tasks   - Eval
           stdOut    : lusaka
           stdOut    : lusaka
     #: MetaStatus (hub): Tasks   - EvalDone
✈  >>> demo.main(pool        = pool.inter(),
                 taskApi     = taskApi,
                 taskApiArgs = taskApiArgs,
                 taskTimeout = taskTimeout)
     #: MetaStatus (hub): Waiting - ForSpokes [ for up to 30 secs ]
     #: MetaStatus (hub): Tasks   - Eval
           stdOut    : lusaka
           stdOut    : lusaka
     #: MetaStatus (hub): Tasks   - EvalDone
✈  >>> demo.main(pool        = pool.interOnePerServer(),
                 taskApi     = taskApi,
                 taskApiArgs = taskApiArgs,
                 taskTimeout = taskTimeout)
     #: MetaStatus (hub): Waiting - ForSpokes [ for up to 30 secs ]
     #: MetaStatus (hub): Tasks   - Eval
           stdOut    : lusaka
           stdOut    : lusaka
     #: MetaStatus (hub): Tasks   - EvalDone
   >>> exit()
   GNU/Linux []
```