

== == == == == == == == == == == == == == == :
**Информатика, вычислительная техника
и управление**

Системный анализ, управление
и обработка информации

*Гагарина Л.Г., доктор технических наук,
профессор, зав. кафедрой
Карпов А.С., аспирант
Нестеров С.А., аспирант
(Московский государственный инсти-
тут электронной техники)*

**МЕТОДОЛОГИЯ ДИАГНОСТИРОВАНИЯ ПРОГРАММНО-АППАРАТНОЙ
СРЕДЫ ИСПОЛНЕНИЯ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ.
ЭЛЕМЕНТЫ УРОВНЕЙ ПАРАЛЛЕЛИЗМА**

В связи с тем, что на сегодняшний день использование технологий параллельных вычислений становится все более важной задачей, крайнюю актуальность приобретают вопросы, связанные с диагностикой таких систем. Немаловажным является разработка методологии диагностирования программно-аппаратных сред информационных систем. Каждая такая среда подразумевает под собой возможность исполнения параллельных программ. Параллельность многопроцессорной программы – это достигнутая ей степень параллельного выполнения, которая в любом случае будет ограничиваться количеством процессоров, физически доступных программе.

При постановке задачи диагностирования системы параллельных вычислений появляется необходимость в определении уровней, в данном конкретном случае, уровней параллелизма и выделения объектов, которые будут подвергаться диагностированию.

Выделяют четыре уровня распараллеливания:

- распараллеливание на уровне задач;
- уровень параллелизма данных;
- уровень распараллеливания алгоритмов;
- параллелизм на уровне инструкций;

Распараллеливание на уровне задач

На данном уровне происходит разбиение одной вычислительной задачи на несколько относительно самостоятельных подзадач, которые можно решить по отдельности. Данный вид распараллеливания прост и в ряде случаев весьма эффективен, правда, в случае с однородной задачей не применим. Что бы распараллелить однородные задачи, нужно спуститься на уровень ниже.

На этом уровне можно выделить ряд элементов, которые можно подвергнуть диагностированию. Это следующие четыре элемента:

- процесс;
- прикладная нить;
- легковесный процесс;
- нить ядра.

В первую очередь понятие задача, подразумевает под собой конкретный процесс операционной системы. В семействе операционных систем Unix каждый процесс выполняется в собственной виртуальной памяти, и, тем самым, процессы защищены один от другого, т.е.