EPOS: Embedded Parallel Operating System

LISHA/UFSC

Prof. Dr. Antônio Augusto Fröhlich
guto@lisha.ufsc.br
http://www.lisha.ufsc.br/~guto

March 2004
EPOS

- Experimental application-oriented operating system
  - Abstractions, scenario aspects, framework, and tools
- High-performance dedicated computing
Tailoring EPOS

application's source code

Analyzer

specified interfaces

Configurator

framework

Info

scenario aspects

frameworks elements

Generator

abstractions and mediators

tailored EPOS
EPOS Framework Metaprogram
EPOS Sample Application: Dinning Philosophers

```
#include <iostream>
#include <synchronizer.h>
#include <thread.h>

using namespace System;
using namespace std;

Synchronizer fork[5];

int philosopher(int n)
{
    int first = (n < 4) ? n : 0;
    int second = (n < 4) ? n + 1 : 4;
    for (; ; )
    {
        cout << "Philosopher " << n
             << " thinking ...\n"
             << "Philosopher " << n
             << " eating ...\n"
             << "Philosopher " << n
             << " thinking ...\n"
             << "Philosopher " << n
             << " eating ...\n"
             << "Philosopher " << n
             << " thinking ...\n"
             << "Philosopher " << n
             << " eating ...\n"
             << "Philosopher " << n
             << " thinking ...\n"
             << "Philosopher " << n
             << " eating ...\n"
             << "Philosopher " << n
             << " thinking ...\n"
             << "Philosopher " << n
             << " eating ...\n"
            
        fork[first].lock();    // get first fork
        fork[second].lock();    // get second fork
        fork[first].unlock();   // release first fork
        fork[second].unlock();  // release second fork
    }
}

int main()
{
    Thread* phil[5];
    for(int i = 0; i < 5; i++)
        phil[i] =
            new Thread(&philosopher, i);
    for(;;);
}
```

Synchronizer {
    constructor(void);
    lock(void);
    unlock(void);
}

Thread {
    constructor(int (*)(int),
               int);
}