FlingOS: Using C# for an OS

# .NET South West – BCS Presentation – 2015-12-02

# Description

FlingOS is an unusual operating system - we use C#. In this talk I'll explain why we chose C#; how it can be more effective for teaching than traditional approaches and I'll walk you through some of our code, including bits our compiler, debugger and main kernel.

# Bio

Edward Nutting is a second year Computer Science and Electronics student at the University of Bristol and founder of FlingOS™ - an educational operating system. Ed has 10 years of experience in programming and education, including teaching at all levels from primary to university. He brings to the table a broad range of knowledge and depth of experience, but with recent emphasis on low level and OS development.

# Notes

## Before we start

* Please to welcome everyone to University of Bristol
* Joint group of .NET South West and BCS
* Roja sends his apologies
* Nearest fire exits
* Recording
* Questions during and after

## Introduction to FlingOS

* What is FlingOS?
* What is the purpose of FlingOS?
* Why is FlingOS needed?
* Why does that make C# appropriate?
* What do we expect of and for students?

## Technical background

* Why C# not Java or similar?
* How powerful is C# compared to C?
* Brief breakdown of compiler structure
* Brief breakdown of OS structure

## Technical details

1. C# to Machine Code : The FlingOS Compiler – Code Generator
   1. MSBuild : C#/VB.Net/F# (and C++?) to IL
   2. FlingOS: IL to ASM (x86 / MIPS / ?)
   3. NASM/GCC: ASM to Machine Code
   4. Ld: Link machine code together
   5. Example: Look at some compiled C# to IL to ASM
   6. Integrated with MSBuild system – nothing manual, errors in the log/window
2. C# to Machine Code : Filling the gaps
   1. C# attributes
   2. Plugging:
      1. Purpose of ASM plugs
      2. Format of a plug file (inc. targeting different architectures)
      3. Example: Look at a plugged method call and plug file
   3. Replacing:
      1. Removal of .NET framework
      2. Alternative approach: Compile without standard library?
      3. Our approach : Filler classes
         1. Objects
         2. Strings
         3. Arrays
3. C# for an OS : Getting Started
4. Example: Before the Main method: Prerequisite plugs and static constructor initialisation
5. Example: The Main method
6. C# for an OS : What about memory management?
   1. The Heap
   2. The Garbage Collector
   3. Example: Our implementations
7. C# for an OS : What about exception handling?
   1. Exception handler methods
   2. Example: Throwing an exception
   3. Example: Handling an exception
   4. Handling exceptions: filters?
   5. Example: Null-reference exception
   6. Example: Index out of range exception
8. FlingOS : What can our system do?
   1. Example: Using the system
   2. Example: Multi-tasking initialisation
   3. Example: IPC using messages
   4. Example: IPC using pipes
   5. Other: PS2, ATA, PCI, USB
9. C# for an OS : What about debugging?
   1. FlingOS Debugger
   2. Alternative: gdb?
   3. Visual Studio integration?
10. FlingOS : What can our system do?
    1. FlingOops testing kernel (MIPS/CI20)
    2. Future plan

## Finishing up

1. Where are we headed?
2. Can you help us out? Support, sponsor or feedback
3. Thanks for listening