Kevin Schaefer
8-20-2014
TNG - The Next Generation OS

Chapter 1

Introduction

This paper documents the system being developed by Kevin Schaefer as a learning exercise in computer science and electrical engineering. It consists of three components:

1. Compiler
2. Computer
3. Emulator

The compiler, and associated programming language, is an exercise in writing a complete language and compiler capable of compiling it into machine code suitable for running on the computer or the emulator. To make it more fun, the language will be themed after Star Trek. The high level programming language that will be developed will be similar to the C or C++ programming languages, but will incorporate specific elements for the computer being developed.

The computer is an exercise in computer or electrical engineering. The computer will be developed with as many low-level electrical components as are feasible. This means that the computer will consist of a large number of transistors and possibly logic ICs to prevent the computer from becoming too large. This basically comes down to the fact that building an XOR gate from transistors is pretty straight forward (and also somewhat busy work), but using ICs that implement XOR gates helps eliminate some of the busy work. On the other hand, certain parts of the computer, namely long-term storage, are going to require something other than basic components.

Finally, the emulator will provide a method of testing the compiler and language without having to copy the output onto the already built computer. This allows for faster development, and is also pretty cool.

Depending on how all of this goes, a converter tool might also be developed to convert code written in the custom language to C++, or vise-versa.

Chapter 2

TNG Architecture

2.1 Introduction

This chapter will provide an in-depth explanation of the architecture of the system being developed.

2.2 Instruction set

This section outlines the instruction set of the computer.

Opcode	Hex	Description
AND	0x1	bitwise AND
OR	0x2	bitwise OR
XOR	0x3	bitwise XOR
INV	0x4	bitwise invert (not)
POP	0x5	pop value off stack
PUSH	0x6	push value onto stack
ADD	0x7	
SUB	0x8	
MLT	0x9	
DIV	0xA	
JPZ	0xB	Jump if zero
JPN	0xC	Jump if negative
LD	0xD	Load data
ST	0xF	Store data

Chapter 3

TNG Language Specification

3.1 File types

Extension	Description
.tng	Source code file
.st	Header file

3.2 Keywords

3.3 Sample program

```
**Section: Credits
  Author: Kevin Schaefer
  Date: 8-20-2014
  Package: Samples
  Email: kevin.l.schaefer1394@gmail.com
**EndSection: Credits
**Section: Code

* This is the main code section in the file

* Define a sub procedure with a parameter of type int, which returns nothing
episode printHello(int param1) space
{
  * Print a message to the screen
  write("Welcome, the parameter is ", param1);

  * Return nothing
  return space;
}

* The main entry point of the program
episode enterprise() space
{
  * Call a sub procedure
  warp printHello(1);
```

```
  * Return nothing
  return space;
}
```

**EndSection: Code