HalOS

Project “HalOS”

By:

Christopher DeGuise

Short Description

- Embedded Operating System targeting the IA-32 platform.

What is an Operating System?

- “Micro” management at its best.
- Provides resource management.
- Provides “applications” developers with a set of tools to work with
- Abstracts the details of the actual hardware.

What an Operating System is not!

- Is not something that you can use as a user.
- Is not the “START” button in Microsoft Windows

Types of Operating Systems

- Microkernel – Smallest and generally real time.
  - QNX
  - uC/OS
- Monolithic – larger and generally not real time.
  - Windows
  - Linux
  - HalOS

Technology/Tools

- C language
- Intel Assembly language
- Tools
  - Linux (Development Platform)
  - GCC (C compiler)
  - NASM (Asm compiler)
  - Doxygen (Documentation generator)
  - Microsoft Visual Source (Version Control)
  - Microsoft Visio (Modeling Tool)

Project Goals

- Small/fast
- Application model similar to uC/OS.
- Provide a system to be used in an educational/learning environment

HalOS Requirements Summary

- Utilize features of the Intel 32bit platform.
- Support a pre-emptive multi-tasking environment.
- Provide an Application Program Interface (API) for software developers.
- Device support
  - Keyboard
  - Video
  - RS-232 (Serial communication)
  - Block Device (Floppy)

Schedule

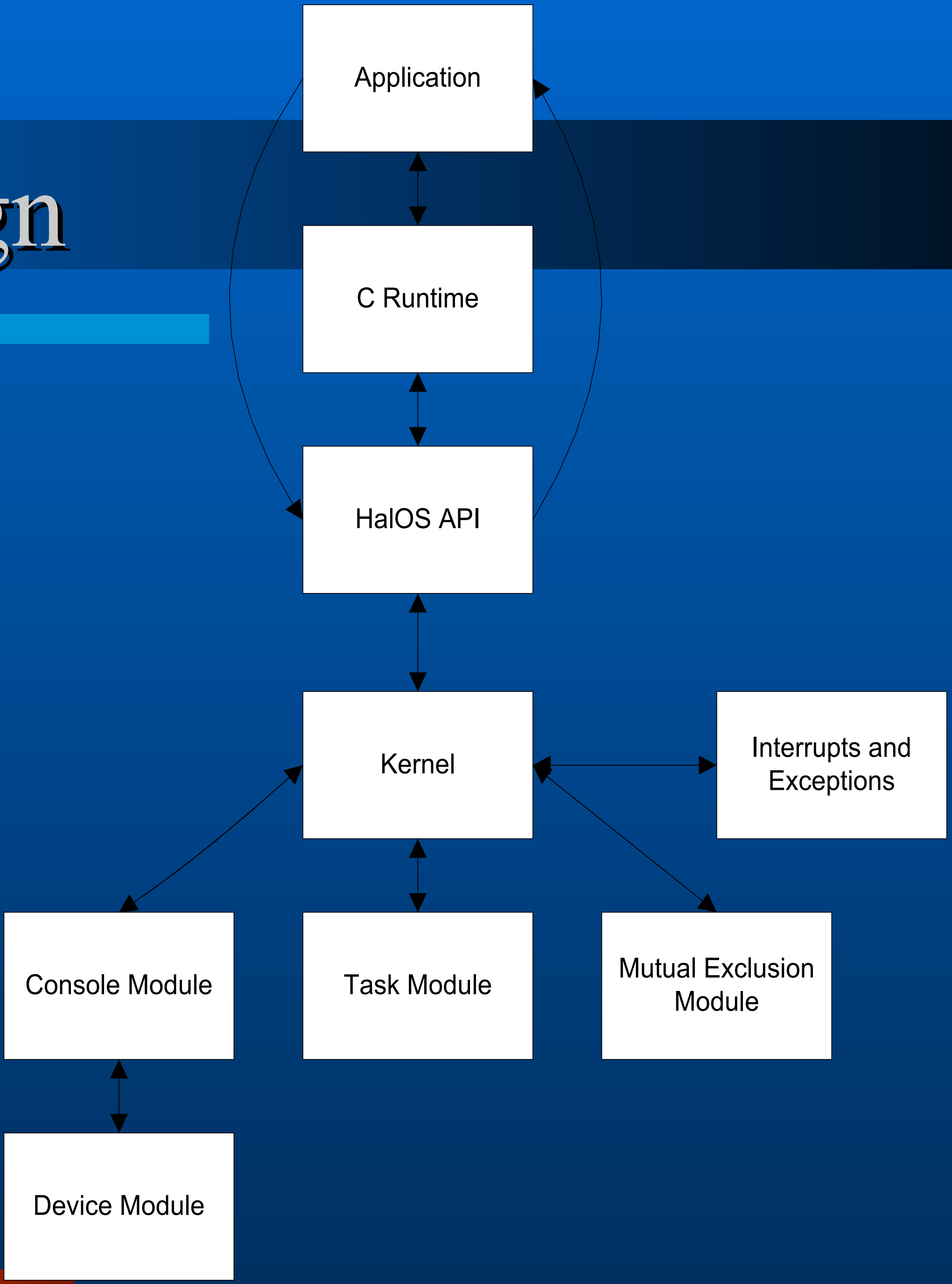
Nov 1	Nov 15	Dec 1	Dec 15	Jan 1	Jan 15	Feb 1		Feb 15	Mar 1	Mar 15		Apr 1	Apr 15	May 1
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			
	🟥	🟥		🟥	🟥		🟥	🟥	🟥		🟥	🟥	🟥	
			🟦			🟦				🟦				

Legend: Sprint Documentation Research

HalOS Components

- Boot Loader
  - First “program” to be executed
  - Puts computer in to a know state
  - Loads Kernel in to memory
- Kernel
  - The actual operating system

System Design



Application

C Runtime

HalOS API

Kernel

Interrupts and Exceptions

Console Module

Task Module

Mutual Exclusion Module

Device Module

Developer Support

- Native Application Program Interfaces.
  - Console
  - Task
  - Mutual Exclusion
  - Keyboard
  - RS-232
- 21 functions ported from C Runtime.
  - Including
    - putchar ,sprintf, etc..
    - memcpy,memset, etc..
    - strlen, strcpy, etc…
    - isdigit, isalpha, etc..

Developer Support (Cont.)

- Kernel object are static. This is for speed reasons.
- Custom build system for the desired system, through the use of #defines.

```
#define TASK_MAX 4
#define CONSOLE_MAX 2
etc..
```

Sample Programs

- Halos API

```
void Task1(void){
    ConsoleWrite(hStdout,”Hello World!, Press any key to
                                                continue.\n”);
    KeyBoardGetCharacter(hStdin);
    TaskClose();
}
```

- C Runtime

```
void Task1(void){
    printf(“Hello World!, Press any key to continue.\n”);
    getchar();
}
```

Must have resources

- News group alt.os.development
- Operating System Design and Implementation (Andrew Tanenbaum)
- Indispensable PC hardware book (Hans-Peter Messmer)

Where to now?

- Implement a memory manager
- VT100 support for terminals

Demo and Q&A