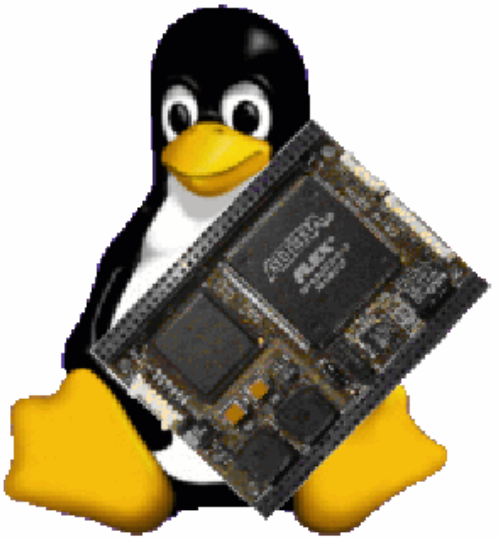
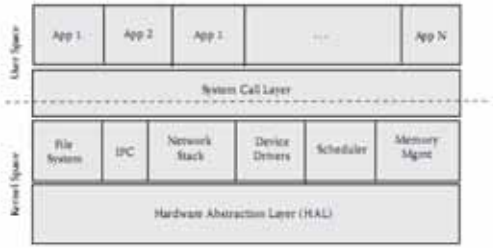
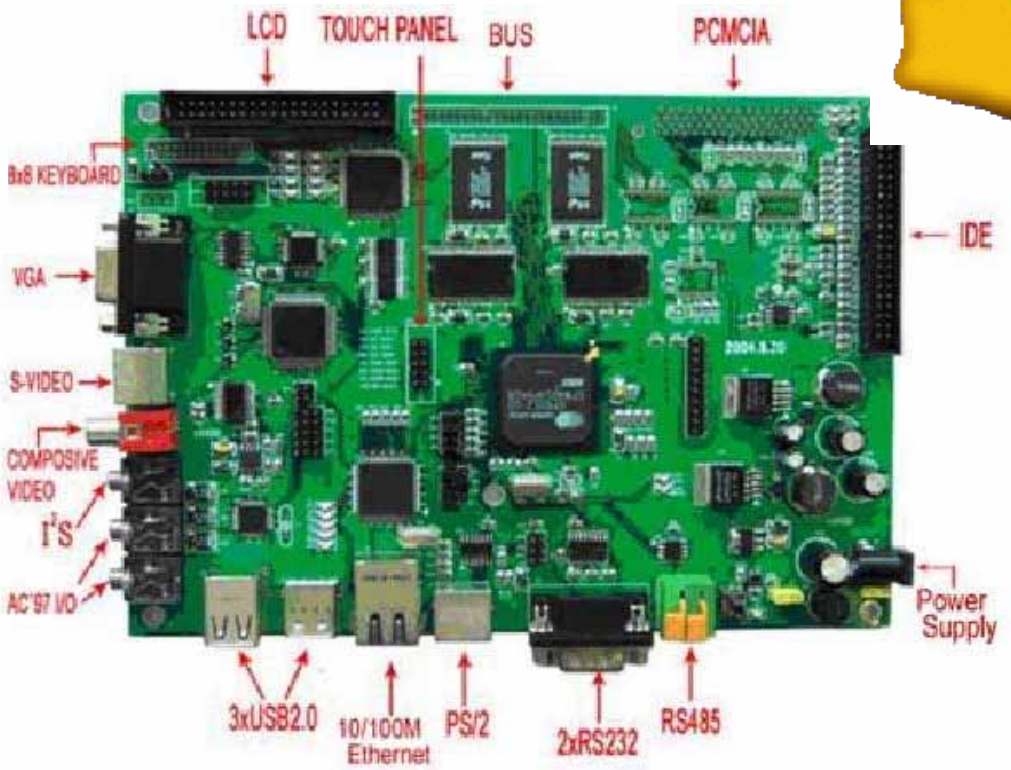
dr inż. Mariusz Kapruziak
mkapruziak@wi.ps.pl
pok. 107, tel. 449 55 44

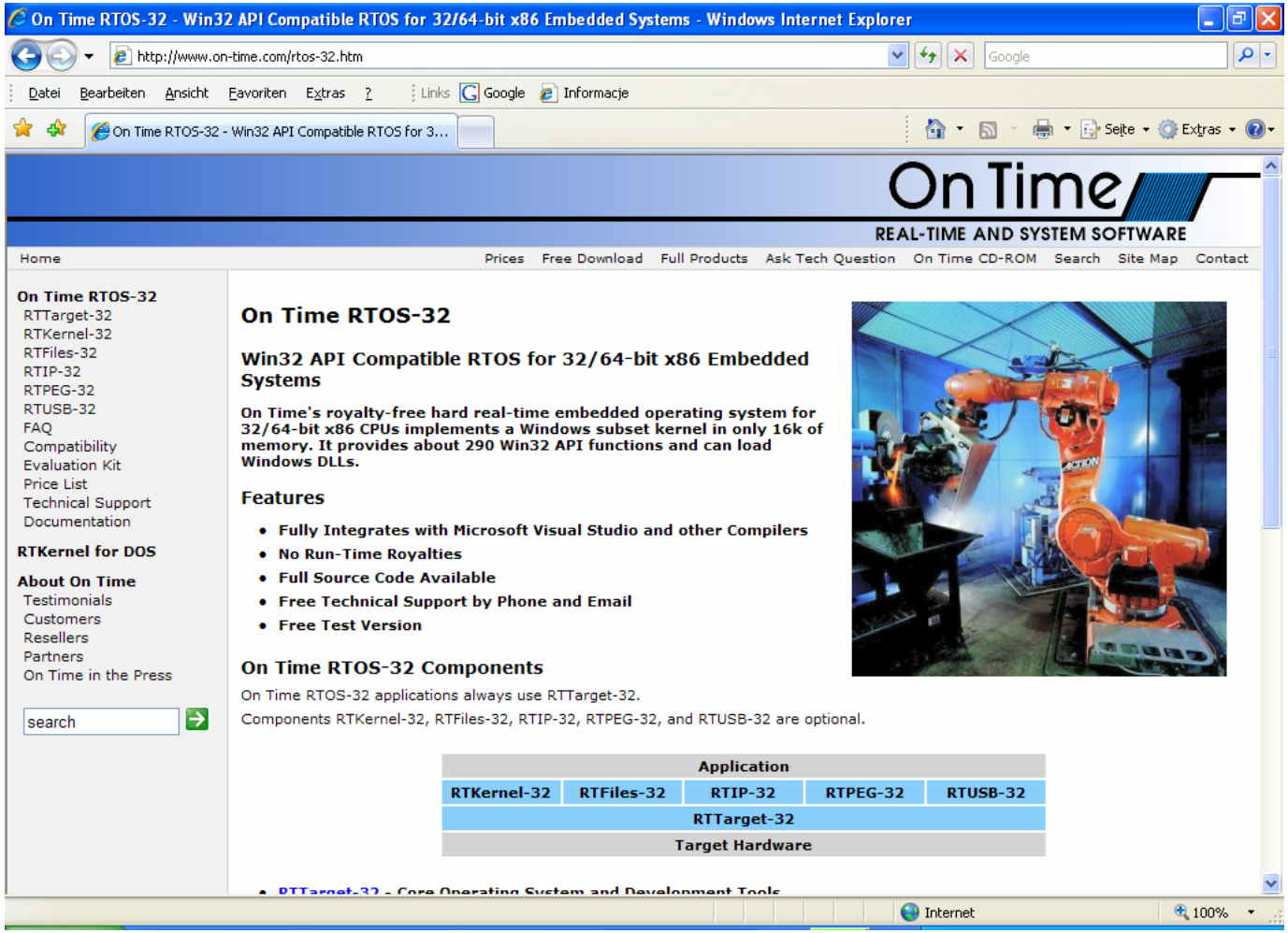
Embedded Linux







Pomysły, pomysły, pomysły....

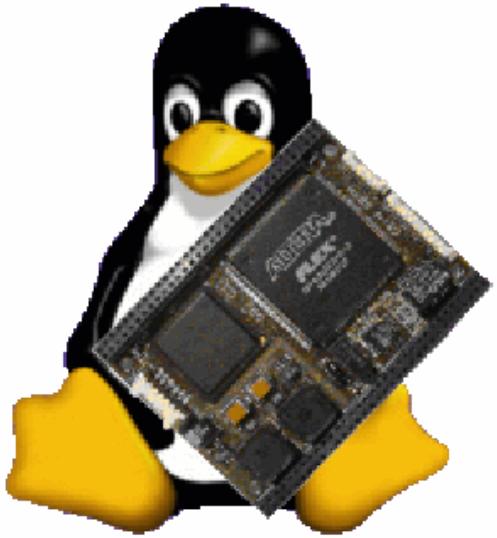


On Time RTOS-32

RT Kernel for DOS

RTUSB-32

Embedded Linux – początki ...



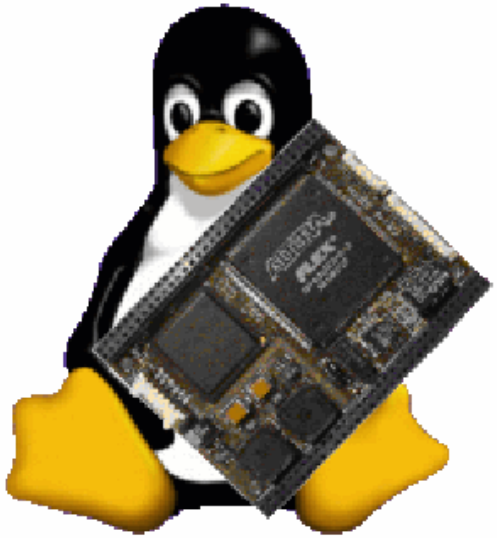
1998 uCLinux

Motorola	DragonBall

1999 ESC = Embedded Systems Conference

Zentropix	RealTimeLinux.org
Lineo	EMLAB (Embedded Advisory Board)
Rick Lehrbaum	Linuxdevices.com
LynuxWorks	BlueCat Linux – Pierwsza komercyjna wersja wbudowanego Linuxa
uCLinux	ARM, ColdFire (Freescale)

Embedded Linux – troche historii



2000 rok

Samsung	Yopy, PDA z Linuxem
Ericsson	HS210, tel. kom. z linuxem
Atmel	AT75C310, chip z obsługą internetu, VoIP, audio na Linuxie
ELC	Embedded Linux Consortium, Rick Lehrbaum, Intel, IBM, ...

Wybrane dystrybucje wbudowanego linuxa

BlueCat Linux – *LynuxWorks, uniwersalna dystrybucja komercyjna, PowerPC, IS-32, ARM, MIPS, x86. Aplikacja VisualLynux.*

Cadenux – *oparta na uCLinux, dla systemów bez MMU.*

DENX – ELDK *(Embedded Linux Development Kit). Zawiera opcje RT. Procesory Xscale, PowerPC, ARM, MIPS x86, SPARC. RTAI (Real-Time Application Interface).*

Embedded Debian (Emdebian) – *okrojony Debian. IA-32, Motorola M68k, SPARC, ARM, PowerPC, MIPS, IA-64*

ELinOS – *SYSGO, dystrybucja komercyjna. Możliwa redukcja zużycia pamięci do 1MB ROM i 2 MB RAM. Dobry manager dostosowujący linux do sprzętu. Aplikacja CODEO dla Eclipse.*

Wybrane dystrybucje wbudowanego linuxa

Metrowerks – *dystrybucja komercyjna wraz z wieloma narzędziami i rozszerzeniami. Procesory x86, ARM, PowerPC, ColdFire. CodeWarrior Development Sudio.*

MontaVista – *komercyjna dystrybucja Linuxa*. W wersji prof. ma wsparcie dla RT. MontaVista DevRocket.

RTLinuxPro – *kernel RT. Architektura dual-kernel: Hard RT/POSIX. Quickboot*

TimeSys Linux – *komercyjna dystrybucja. Windows/Linux cross-development. Przeciętna jak się wydaje.*

Problemy wykorzystania Linuxa w systemach wbudowanych

Wielkość systemu:

konfiguracja systemu. Minimaly Linux 4MB SDRAM i 2MB Flash. Mniejsze dystrybucje to: uxLinux, ELKS (Embedded Linux Kernel Subset), ThinLinux.

Linux jako system Real Time

Preemptive kernel, RT-capable scheduler. Dual kernel.

Linux i systemy komercyjne

GPL (General Public License), UWAGA !!!
LGPL (Lesser GPL) – można nie publikować własnego kodu

Komercyjne dystrybucje Linuxa/darmowe dystrybucje

Efekt amatorskości/koszty, przywiązanie się do egzotycznej dystrybucji

Licencja GPL

1) Aplikacje tylko używające system Linux poprzez zwykłe wywoływanie funkcji systemowych nie muszą mieć licencji GPL

> *This copyright does "not" cover user programs that use kernel services by normal system calls - this is merely considered normal use of the kernel, and does "not" fall under the heading of "derived work."* [2]

2) Fragmenty wzbogacające kernel mogą EWENTUALNIE NIE MIEĆ licencji GPL tylko jeśli są dołączane do kernela jako moduły.

3) Patche i modyfikacje kernela nie dołączane dynamicznie jako moduły MUSZA BYĆ GPL.

4) WYKORZYSTANIE BIBLIOTEK/APLIKACJI Z LICENCJA GPL TAKŻE SKUTKUJE WYMUSZENIEM LICENCJI GPL.

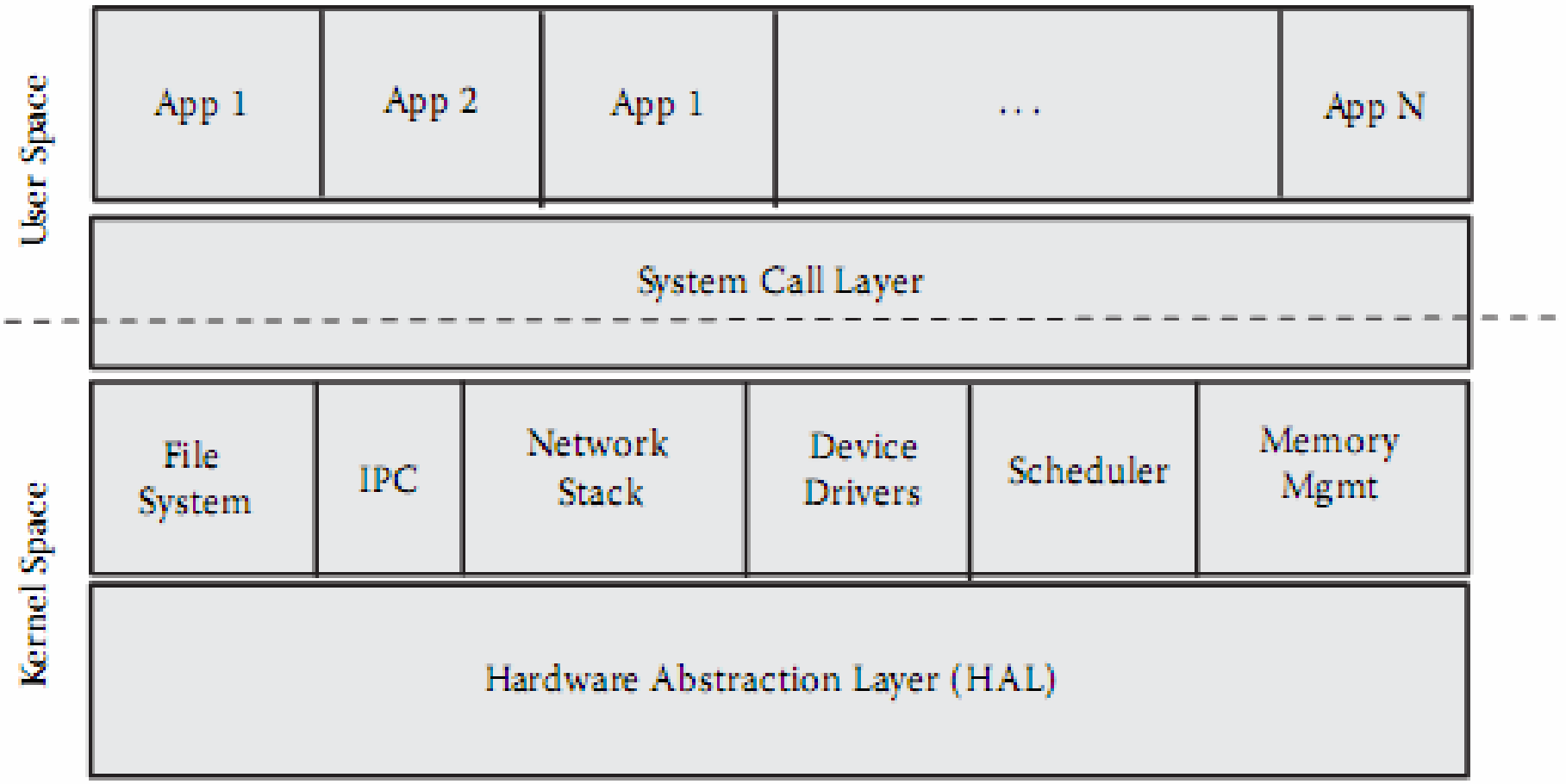
5) Biblioteki systemowe (jak libc, pthreads itp.) mają licencje LGPL i nie wiążą się z koniecznością posiadania licencji GPL na kod je wykorzystujący.

Embedded Linux dnia dzisiejszego

Własne przemyślenia dowolne:

- Kernel dobry i stabilny
- Troche wolne i za duże
- Nie wiadomo po co jest wymuszenie systemu plików
- Amatorskie ☹ - interfejs użytkownika
- Chyba ma duża przyszłość, nie powinno się ignorować
- W sumie dużo dystrybucji, nieuporządkowany rynek
- Każda płytka ma swoją dystrybucje
- Dość kiepskie narzędzia konfiguracji – szczególnie dla ToolChain

Linux Kernel



HAL (Hardware Abstraction Layer)

BSP (Board Support Package) w innych RTOS

HAL w Linux nie ma standardowego API

Procesory wspierane przez Linux HAL:

- MIPS
- PowerPC
- ARM
- M68K
- CRIS
- V850
- SuperH

Memory Manager, File System

Pages, 4KB

Osobno traktowane strony aplikacji i kernela

VFS – Virtual File System

System plików jest obowiązkowy – root file system

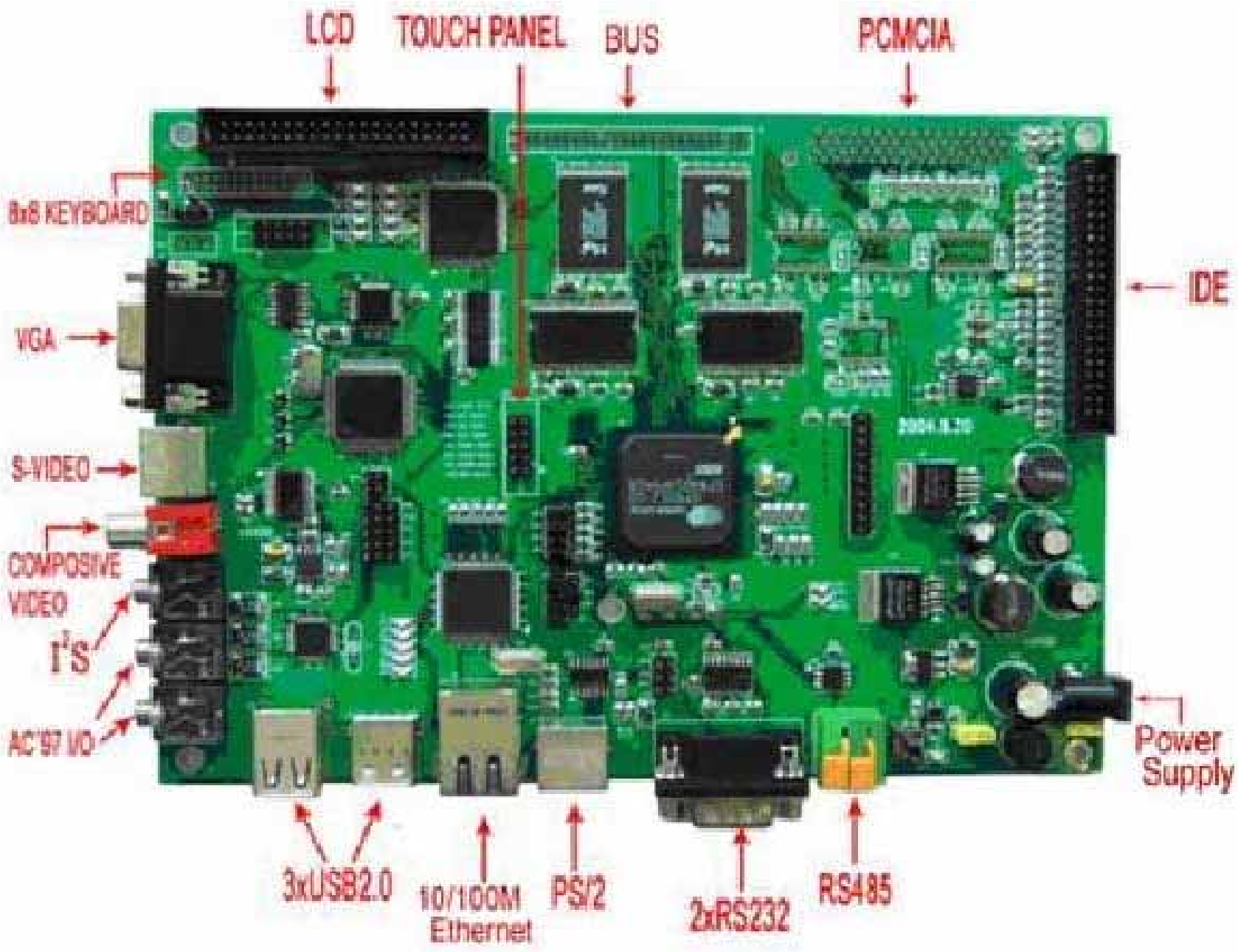
Typy systemów plików:

- EXT2 – klasyczny system plików
- ROMFS – Read Only File System
- RAMFS – Memory-based File System
- DEVFS – Pseudo file system for device files
- JFFS2 – Obsługa dedykowana dla pamięci Flash

Start systemu Linux

1. **Faza BootLoadera** – inicjalizacja sprzętu oraz przekazanie kontroli do systemu
2. **Start jądra**:
    1. Kernel_entry / stext – procedura w assemblerze inicjalizująca start jądra. Znajduje się zazwyczaj w: kernel/head.s
    2. Start_kernel(), orginalnie w pliku init/main.c – inicjalizacja i uruchomienie wątku „idle” (process id 0)
    3. wywołanie funkcji setup_arch(), inicjalizacja wybranych zasobów zależnych od platformy jak: rozpoznanie CPU, zidentyfikowanie ramdisk, inicjalizacja stronicowania
    4. trap_init() – inicjalizacja procedur obsługi sytuacji wyjątkowych
    5. init_IRQ() – inicjalizacja kontrolera przerwań
    6. Time_init – inicjalizacja timerów
    7. Console_init() – inicjalizacja urządzenia szeregowego do obsługi konsoli
    8. Calibrate_delay() – kalibracja dla funkcji udelay()
    9. Inicjalizacja dodatkowych sterowników urządzeń
3. **Inicjalizacja przestrzenii użytkownika:**
    1. Inicjalizacja i przekazanie sterowania do procesu init

Embest NK9315



Embest NK9315 - sprzęt

Hardware specifications of the NK9312/9315 development board as below:

Dimensions: 140x200mm

Processor : Cirrus Logic EP9312/9315 based on 32-bit ARM920T core microcontrollers

Power input : DC12.0V

32M NOR Flash Memory

64M 32bit SDRAM

12 CHANNELS DMA

Optional LCD Display and Touch-Screen interface

Analog VGA connection

Composite Video and S-Video Output Connections

IDE Interface

Three Port USB Host

20pin standard JTAG interface

Real-Time Clock

I2S Interface

AC97 Module

8*8 Keyboard interface

PS/2 Interface

3 UARTs. One is RS485, and the other two are RS232 , share with the same DB-9 connector

Embest NK9315 - LINUX

Linux Kernel 2.4.21

Opcje standardowe

1) Frame buffer driver, support these CRT:
640X480X8, 640X480X16, NK9315 LCD, NK9315 TV/OUT
2) Touchscreen driver
3) Internet driver
4) PS/2 keyboard driver
5) Serial Terminal driver
6) I2S sound driver
7) AC97 sound driver
8) RAMDISK driver

Opcje możliwe do dodania

The following function is not compiled in the default kernel:

1) USB driver
2) IDE driver
3) CD-ROM and DVD-ROM driver
4) SCSI driver
5) NFS
6) DEV-FS
7) CRAMS
8) JFFS2

eCOS + RedBoot

eCOS – system RT o wysokiej efektywności.

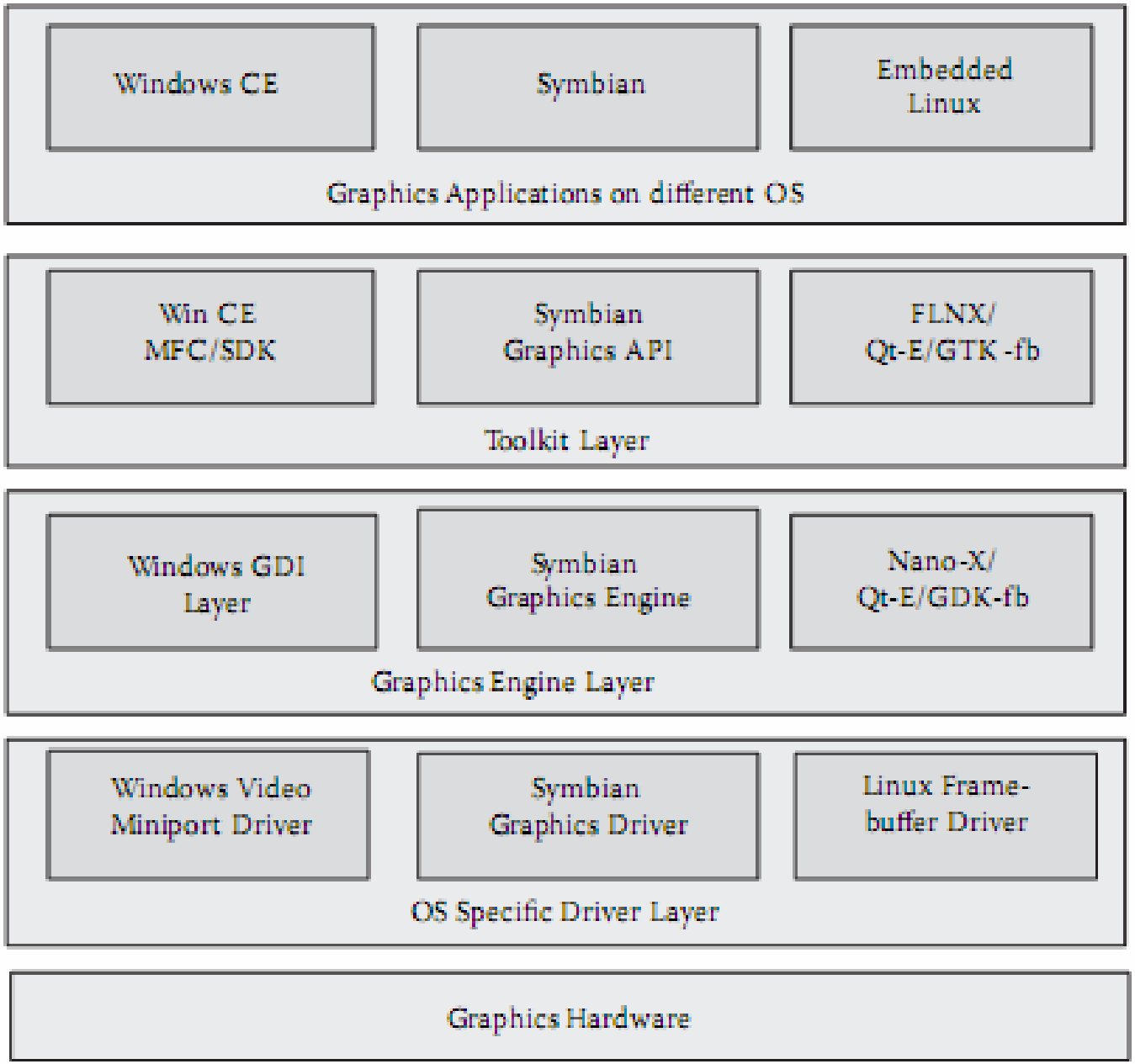
- Hardware Abstraction Layer (HAL)
- Real-time kernel
  - Interrupt handling
  - Exception handling
  - Choice of schedulers
  - Thread support
  - Rich set of synchronization primitives
  - Timers, counters and alarms
  - Choice of memory allocators
  - Debug and instrumentation support
- POSIX compatible API
- ISO C and math libraries
- Serial, ethernet, wallclock and watchdog device drivers
- USB slave support
- TCP/IP networking stacks
- GDB debug support

REDBOOT – bootloader, w oparciu o eCOS HAL. Umożliwia załadowanie do pamięci danych za pomocą portu szeregowego lub ethernetu. Umożliwia zapis do pamięci FLASH.

RedBoot – mapa pamięci

Name	FLASH addr	Mem addr	Length	Entry point
RedBoot	0x60000000	0x60000000	0x00040000	0x00000000
RedBoot config	0x61F80000	0x61F80000	0x00001000	0x00000000
FIS directory	0x61FC0000	0x61FC0000	0x00040000	0x00000000
ramdisk	0x60040000	0x01000000	0x00180000	0x01000000
vmlinux	0x601C0000	0x00218000	0x00140000	0x00218000

Aplikacje graficzne



Aplikacje graficzne - tabela

Name	*License*	*Comments*
Nano-X www.microwindows.org	GPL/MPL	Windowing environment providing Win32- and X11–like APIs, targeting embedded systems.
FLNX www.fltk.org	LGPL	FLTK toolkit ported over Microwindows
MiniGUI www.minigui.com	LGPL	Compact graphic user interface support system for Linux. MiniGUI defines some Win32-like APIs for the applications; provides a small windowing system support library
DirectFB www.directfb.org	LGPL	A thin library that provides hardware graphics acceleration, input device handling and abstraction, integrated windowing system with support for translucent windows, and multiple display layers
PicoGUI www.picogui.org	LGPL	A new graphic user interface architecture designed with embedded systems in mind; includes low-level graphics and input, widgets, themes, layout, font rendering, network transparency
Qt/Embedded *www.trolltech.com/products/embedded/index.html*	QPL GPL	C++-based windowing system for embedded devices, provides most of the Qt API
GTK+/FB www.gtk.org	LGPL	Frame buffer port of the popular GTK+ windowing system.

Polecana literatura

1) P. Raghavan, A. Lad, S. Neelakandan, *Embedded Linux System Design and Development*, Auerbach Publications 2006

2) D. Bovet, *Understanding the Linux Kernel*, O'Reilly 2005

dr inż. Mariusz Kapruziak
mkapruziak@wi.ps.pl
pok. 107, tel. 449 55 44

KONIEC

Propozycje tematów

Architektura Komputerów:

1) Implementacja procesora z asynchroniczną ścieżka danych w FPGA

2) System dydaktyczny do prezentacji działania architektury ARM

3) Implementacja procesora 8086 w FPGA

4) System operacyjny czasu rzeczywistego do obliczeń elastycznych w oparciu o algorytmy planowania Schwarzfischer'a

Propozycje tematów

Dedykowane organizacje procesorów

5) Szacowanie kosztu implementacji algorytmów o zmiennej strukturze

6) Aplikacja do rozlokowania na sprzęt i oprogramowanie dla wielu algorytmów jednocześnie z wykorzystaniem algorytmu GCLP-MF

7) Optymalizacja powierzchnii układu poprzez użycie algorytmów zwijania

8) Środowisko generowania kodu dla FPGA do zdarzeniowych aplikacji graficznych

Propozycje tematów

Aplikacje systemów wbudowanych

9) System do lokalizacji intruza w pomieszczeniu na podstawie sieci mikrofonów

10) Lokalizacja i śledzenie obiektu w torze wizyjnym

11) Analizator stanów logicznych na FPGA

12) Robot do walk sumo

13) Roboty do gry zespołowej w piłkę

14) System zdalnej obserwacji i zarządzania obiektem.

15) Implementacja komunikacji USB jako IPCore w FPGA

Propozycje tematów

Widzenie maszynowe

16) Algorytm lokalizacji w przestrzenii robota na podstawie toru wizyjnego

17) Algorytm i implementacja w FPGA korekcji sygnału z czujników na bazie algorytmów statystycznego przetwarzania sygnałów.

18) Rozpoznawanie defektów w aplikacji inspekcji maszynowej

19) Implementacja algorytmów detekcji ruchu w FPGA

20) Implementacja transformaty Hough w FPGA

Propozycje tematów

DSP i systemy komunikacyjne (w szczególnośći komunikacji radiowej)

21) Algorytm elastycznej transformaty DFT w FPGA

22) Algorytm elastycznego filtru Butterwortha w FPGA

23) Modem do transmisji szerokopasmowej w oparci o techniki nierównomiernego próbkowania

24) Implementacja elastycznej konwersji częstotliwości

25) Implementacja filtracji typu „particle filter”

26) Metoda automatycznego doboru szerokości bitowej na różnych etapach przetarzania algorytmu DSP.

Propozycje tematów

Systemy automatyczne i inne (luźne):

27) Implementacja algorytmów uczenia maszynowego w FPGA

28) Implementacja i sprawdzenie teorii map lokacyjnych Treisman'a

29) Implementacja i sprawdzenie teorii proceduralnej reprezentacji informacji

30) Wpływ mechanizmów elastycznego sterowania algorytmem estymacji na oszacowanie parametrów estymatora za pomocą CRLB.