RetroBSD - a minimalistic Unix system

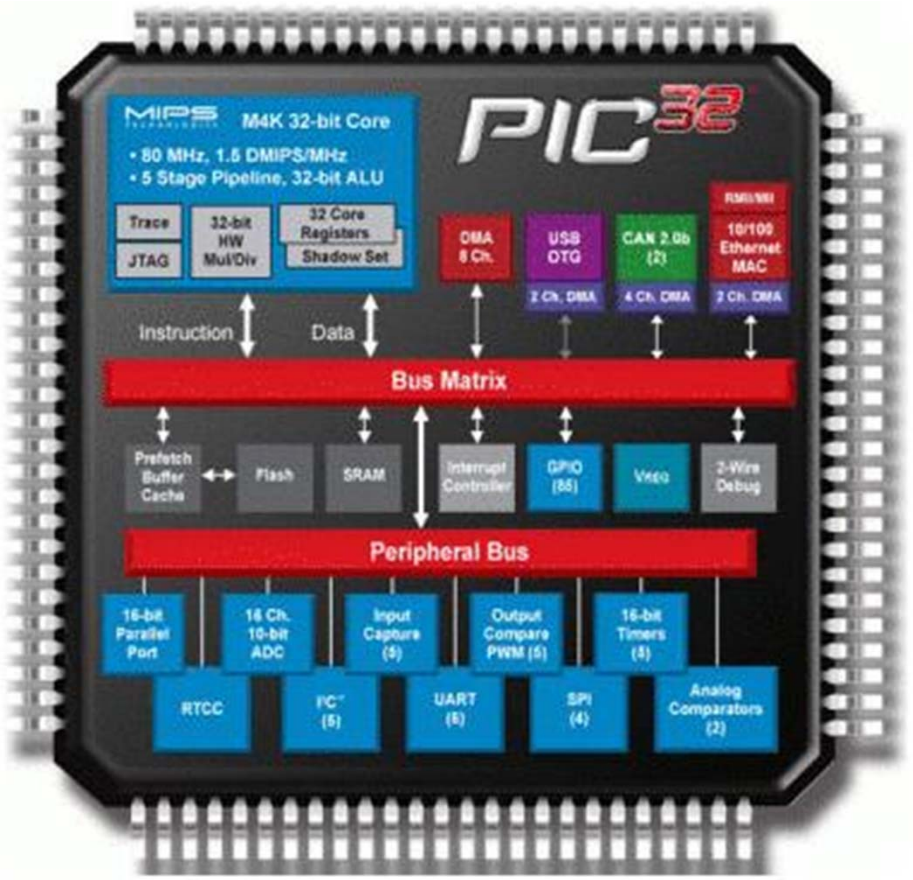
Igor Mokoš
pito@volna.cz
bsd_day
Bratislava
5.11.2011

RetroBSD

- RetroBSD is a port of 2.11BSD Unix intended for small embedded systems
- Currently Microchip PIC32MX 32bit microcontroller with at least 128 Kbytes of RAM and 512 Kbytes of Flash is supported
- Developer - Serge Vakulenko
- Home page http://retrobsd.org/

RetroBSD - MCU supported

- MIPS M4K architecture, with executable data memory and flexible RAM partitioning between user and kernel modes
- RAM 128kbyte (!)
- Flash 512(+12)kbyte
- Fcpu 80MHz, 80mips
- PIC32MX695F512L(100pin)
- PIC32MX795F512L(100pin)
- PIC32MX695F512H(64pin)
- PIC32MX795F512H(64pin)

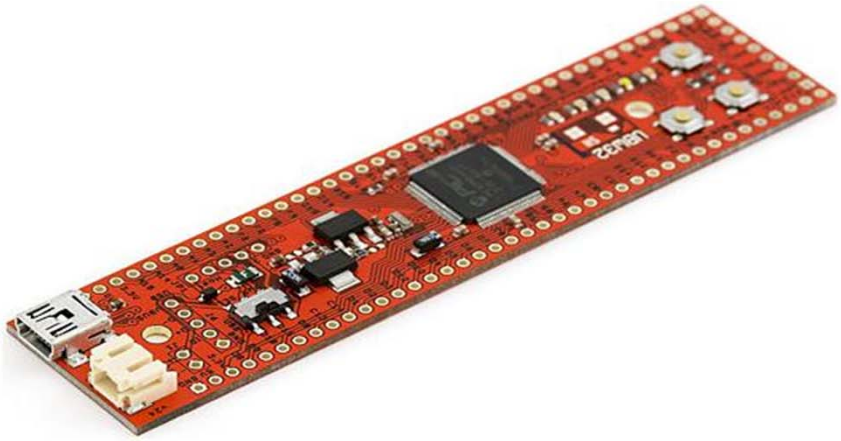


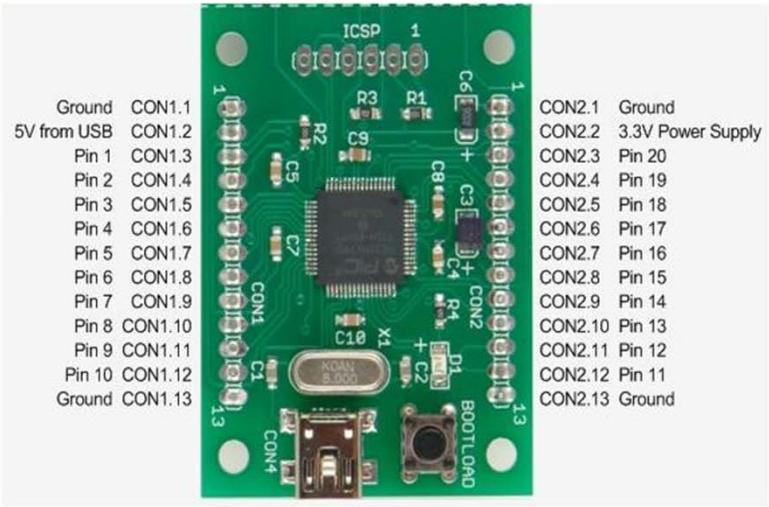
RetroBSD - Minimalistic system

- **“A Minimalistic system”**
  - 1x MCU
  - 1x SD-card
  - 4x LEDs
  - 3.3V voltage regulator
  - 10 resistors, 10 capacitors, breakout board
  - 8MHz crystal
  - Serial connectivity via RS232, Bluetooth, USB
  - Battery

RetroBSD - Minimalistic

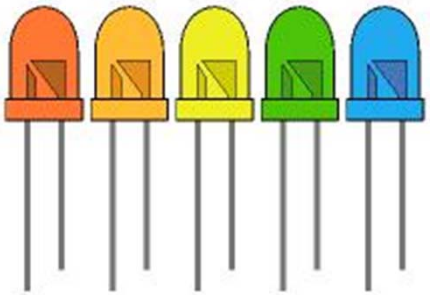
- Supported Boards:





RetroBSD – a typical mini-system

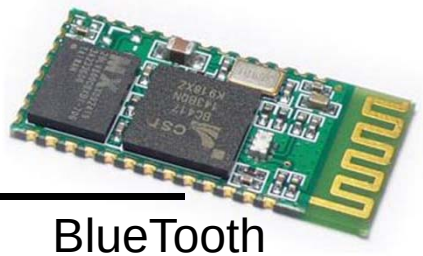
KERNEL
I/O
DISK
SWAP
POWER



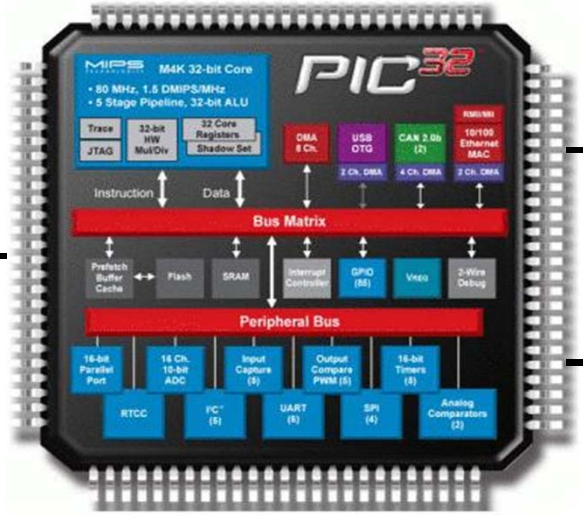
BSD
Filesystem
Swap

16Mb-16Gb

SPI



BlueTooth
(optional)



GPIO

Serial
Console

UART

USB

Bootloader
Serial
Power

RetroBSD – major changes against 2.11BSD

Serge comments on RetroBSD development:

* Kernel sources fixed to compile for MIPS using GCC. Added function prototypes. Fixed all gcc -Wall warnings..

* Removed networking layer: no chances to fit into available RAM. It's easy to add them back later..

* Removed accounting and kernel options QUOTA, EXTERNALITIMES, SOFUB_MAP..

* Removed PDP-dependent features: memory map, clicks, floating point stuff, overlays, separate code/data space, unibus, UCB_CLISTS, drivers..

* Removed useless syscalls: getlogin/setlogin, phys, fperr..

RetroBSD – major changes against 2.11BSD

* Removed struct text. We have not enough incore memory to store shared executable code..

* All system data structures fixed to use word-sized data types. Short and char values are not efficient for RISC architecture..

* Block size increased from 512 bytes to 1 kbyte..

* Filesystem structure changed for 1-kbyte block size..

* Implemented single-process user memory allocation. No malloc/mfree for core memory: all allocation is static..

RetroBSD – major changes against 2.11BSD

* Implemented setjmp/longjmp.. There will be two U areas allocated: one for swapper (i.e. process 0) and another for all other processes. Switching is performed in longjmp, by copying the data

* Implemented swapping to file on a root filesystem.. A swap file is a special contigious region, created by mkfs

* Implemented PIC32 simulator, based on VirtualMIPS engine..

* It was a key question: is it possible to implement process switching without MMU?.. Kernel stack is a part of U area, it should remain at fixed address, to be used by interrupt handlers. But it must be replaced when switched to a new process. If occured, that there is a single place, where it could be implemented: longjmp() routine. In 4.3BSD it is called resume(). It acts much like the standard longjmp() jumping to previously saved thread context, but before it a U area is "remapped", resulting in a process switch.

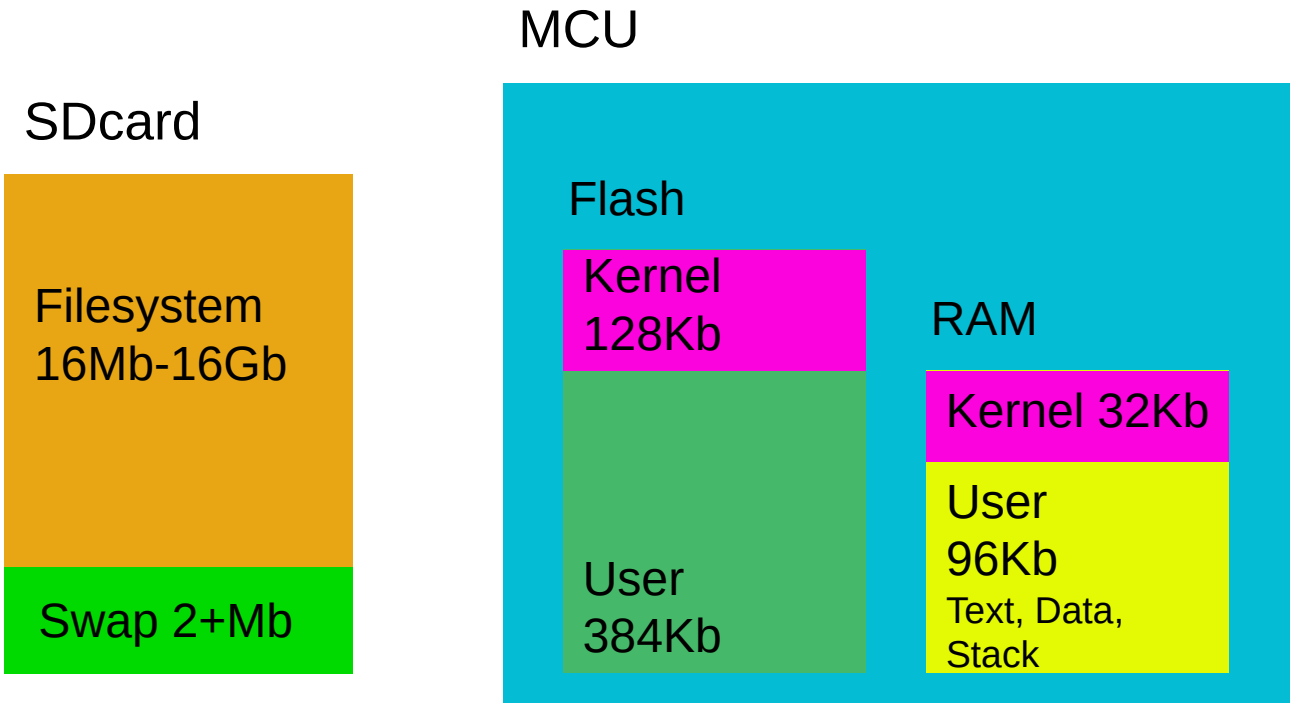
RetroBSD – major changes against 2.11BSD

* "Remapping" is performed by exchanging a contents of U area..

  Actually, there are two areas: U and U0, allocated as static arrays at fixed addresses (in a linker script). We explore the feature, that a process switch could happen only from user program to swapper (process 0), or from swapper to a user program. When a user process is running, a U area contains it's struct user and a kernel stack. At this time, U0 area contains a copy of U data for swapper. When we switch to swapper, the contents of U and U0 is exchanged. Next we switch from swapper to another user process, and they are exchanged again. When a swapper wants to change a process, it saves the user code, data and U0 area to disk, reads another process' data from disk and calls longjmp().

* The major obstacle was a debugging of kernel.. Implementing context switching, interrupt handlers, masking/unmasking, user mode signals is a hard task. Developing a simulator was also an interesting and tricky part.

RetroBSD – memory maps



Sdcard:
Write 250-600Kbytes/sec
Read 600-1700Kbytes/sec

12.5ns RAM access

RetroBSD – dev tools

- GCC 4.6.1 for MIPS
- http://retrobsd.googlecode.com/svn/trunk/ retrobsd
- 2.11BSD-tape utilities (optimized for size)
- cpp-chipKIT-cxx 4.5.1

The following components are built:
----
crt0: C startup routine
libc: C library
elf2aout: utility for converting ELF files to BSD a.out format
fsutil: utility for creating BSD filesystem images
virtualmips: MIPS simulator, with PIC32 extensions
unix.hex: BSD kernel, to be flashed to PIC32
bin/* utilities
sbin/* utilities
root.bin: image of root filesystem, to be copied to SDcard

RetroBSD – today's limits

- PIC32MX MIPS only
- 128Kbytes RAM = 32Kb Kernel + 96Kb User
- Max 20 (10) tasks/jobs in timesharing
- Size of the User's job max 96Kb (code, data, stack)
- 1x Serial UART (up to 6) + 1x Serial via USB
- 1x SD-card (up to 4 SPIs, ~unlimited)
- SWAP on the SDcard ("slow")

RetroBSD – To Do

- A very long To Do list
- USB stack (console/serial)
- Multiport UART
- Multicard support
- TCP/IP stack
- SDRAM for SWAP (?)
- Tools, Applications, etc.
- ARM (?)

Contributors wanted !

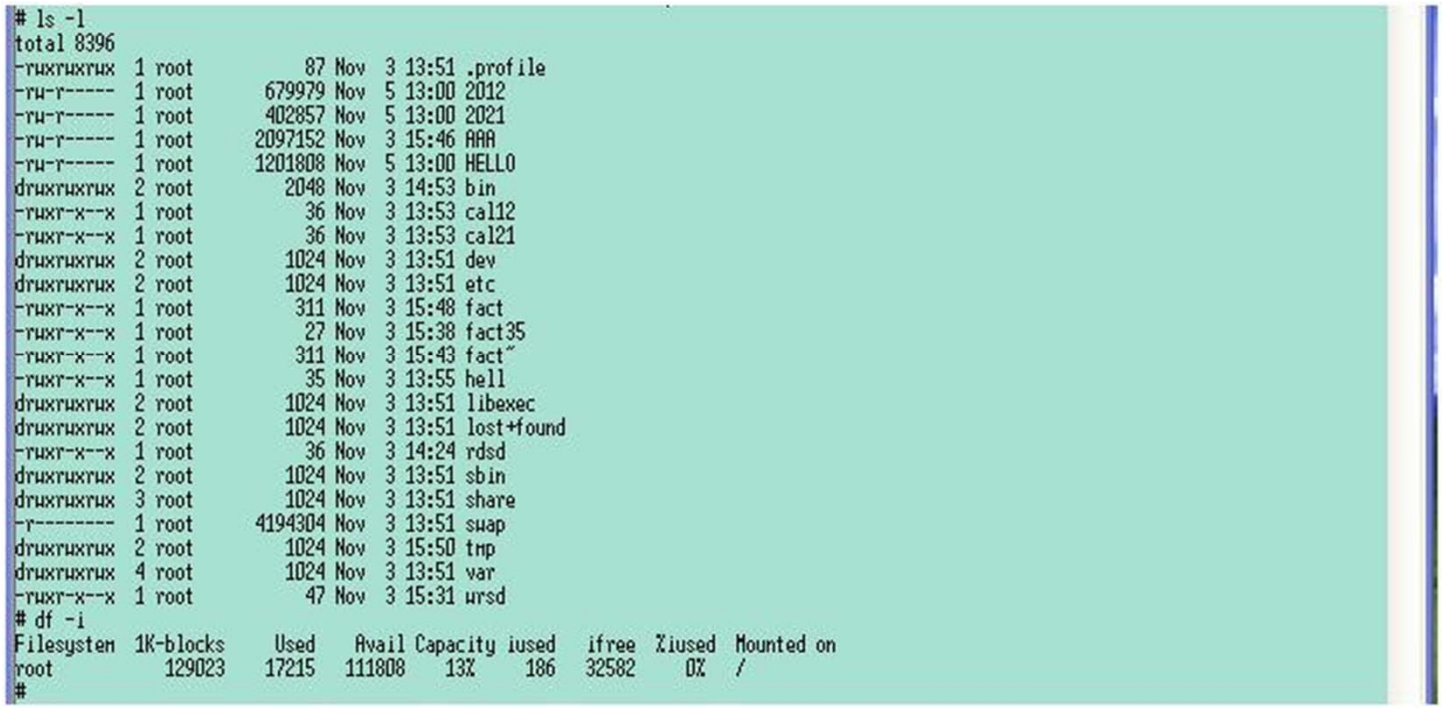
RetroBSD – Demonstration

COM17:115200baud - Tera Term VT

File Edit Setup Control Window Help

```
2.11 BSD Unix for PIC32, revision #308:
    Compiled 2011-10-08 by pito@ubuntu:
    /home/pito/retrobsd/sys/pic32/ubu32
phys mem  = 128 kbytes
user mem  = 96 kbytes
root dev  = (0,0)
console: port UART2
sd0: port SPI2, select pin E4
sd0: type SDHC, size 3872256 kbytes, speed 13 Mbit/sec
root size = 131072 kbytes
swap size = 4096 kbytes
erase, kill ^U, intr ^C
# date 1111051300
Sat Nov  5 13:00:00 PST 2011
# sh cal21 &
5
# sh cal12 &
35
# ps alx
  F S   UID   PID  PPID CPU PRI NICE  ADDR  SZ WCHAN    TTY TIME COMMAND
  3 S     0     0     0  41   0   0 0x6800   3 runin    ?   0:07 swapper
200 S     0     1     0   1  30   0  0x84a  36 proc     ?   0:01 init -s
  1 Z     0    48     5   0   0   0 0x6800  20          ?   0:01 <defunct>
200 S     0     2     1   0  30   0  0x816  57 proc     co  0:02  (sh)
200 R     0     5     2   0  30   0  0x81c  57          co  0:03 sh cal21
200 S     0    35     2   0  30   0  0x810  57 proc     co  0:02 sh cal12
 10 R     0    50    35   1   0   0  0x804  57          co  0:02 sh cal12
  1 R     0    51     2   3  50   0 0x7400  46          co  0:01 ps alx
# sh hell &
111
# ps alx
  F S   UID   PID  PPID CPU PRI NICE  ADDR  SZ WCHAN    TTY TIME COMMAND
  3 S     0     0     0  47   0   0 0x6800   3 runin    ?   0:18 swapper
200 S     0     1     0   1  30   0  0x84a  36 proc     ?   0:01 init -s
  1 Z     0   122    35   0   0   0 0x6800  20          ?   0:01 <defunct>
  1 Z     0   123     5   0   0   0 0x6800  56          ?   0:01 <defunct>
200 S     0     2     1   0  30   0  0x81c  57 proc     co  0:02  (sh)
200 R     0     5     2   0  30   0  0x9bc  57          co  0:04 sh cal21
200 R     0    35     2   0  30   0  0x816  57          co  0:05 sh cal12
200 S     0   111     2   0  30   0  0x80a  57 proc     co  0:01 sh hell
 10 R     0   124   111   0   0   0  0x804  57          co  0:01 sh hell
  1 R     0   125     2   2  50   0 0x7400  46          co  0:01 ps alx
```

RetroBSD – Demonstration



```
# ls -l
total 8396
-ruxruxrux  1 root          87 Nov  3 13:51 .profile
-ru-r-----  1 root      679979 Nov  5 13:00 2012
-ru-r-----  1 root      402857 Nov  5 13:00 2021
-ru-r-----  1 root     2097152 Nov  3 15:46 AAA
-ru-r-----  1 root     1201808 Nov  5 13:00 HELLO
druxruxrux  2 root        2048 Nov  3 14:53 bin
-ruxr-x--x  1 root          36 Nov  3 13:53 cal12
-ruxr-x--x  1 root          36 Nov  3 13:53 cal21
druxruxrux  2 root        1024 Nov  3 13:51 dev
druxruxrux  2 root        1024 Nov  3 13:51 etc
-ruxr-x--x  1 root         311 Nov  3 15:48 fact
-ruxr-x--x  1 root          27 Nov  3 15:38 fact35
-ruxr-x--x  1 root         311 Nov  3 15:43 fact~
-ruxr-x--x  1 root          35 Nov  3 13:55 hell
druxruxrux  2 root        1024 Nov  3 13:51 libexec
druxruxrux  2 root        1024 Nov  3 13:51 lost+found
-ruxr-x--x  1 root          36 Nov  3 14:24 rdsd
druxruxrux  2 root        1024 Nov  3 13:51 sbin
druxruxrux  3 root        1024 Nov  3 13:51 share
-r--------  1 root     4194304 Nov  3 13:51 swap
druxruxrux  2 root        1024 Nov  3 15:50 tmp
druxruxrux  4 root        1024 Nov  3 13:51 var
-ruxr-x--x  1 root          47 Nov  3 15:31 urstd
# df -i
Filesystem  1K-blocks     Used    Avail Capacity iused   ifree  %iused  Mounted on
root           129023    17215   111808    13%     186   32582     0%   /
#
```

RetroBSD – Demonstration


```
# cat fact
50 k
48 [d1-d1<F*]dsFxp
14 [d1-d1<F*]dsFxp
16 [d1-d1<F*]dsFxp
18 [d1-d1<F*]dsFxp
20 [d1-d1<F*]dsFxp
22 [d1-d1<F*]dsFxp
24 [d1-d1<F*]dsFxp
26 [d1-d1<F*]dsFxp
28 [d1-d1<F*]dsFxp
30 [d1-d1<F*]dsFxp
32 [d1-d1<F*]dsFxp
36 [d1-d1<F*]dsFxp
38 [d1-d1<F*]dsFxp
40 [d1-d1<F*]dsFxp
42 [d1-d1<F*]dsFxp
44 [d1-d1<F*]dsFxp
q
# dc fact
12413915592536072670862289047373375038521486354677760000000000
87178291200
20922789888000
6402373705728000
2432902008176640000
112400072777607680000
6204484017332394393600000
403291461126605635584000000
30488834461171386050150400000000
2652528598121910586363084800000000
263130836933693530167218012160000000
37199332678990121746799944815083520000000000
5230226174666011117600072241000742912000000000
815915283247897734345611269596115894272000000000
14050061177528798985431426062445115699363840000000000
2658271574788448768043625811014615890319638528000000000
# 
```

RetroBSD – Demonstration


```
# cd bin
# ls
HELLLO    comm      false     iostat    mv        rev       su        tsort
apropos   cp        fgrep     join      nice      rm        sum       tty
basename  date      file      kill      nm        rmail     sync      uname
bc        dc        find      la        nohup     rmdir     sysctl    uniq
cal       dc.core   grep      last      od        rz        sz        vmstat
cat       dd        groups    ln        pagesize  sed       tail      w
cb        df        head      login     passwd    sh        tar       wall
chflags   diff      hello     ls        pr        size      tee       wc
chgrp     du        hello.c   mail      printf    sleep     test      whereis
chmod     echo      hello.c~  man       ps        sort      time      who
chpass    ed        hostid    mesg      pwd       split     touch     whoami
cmp       egrep     hostname  mkdir     re        strip     tr        write
col       expr      id        more      renice    stty      true      xargs
# re hello.c
```

```
#include <stdio.h>

int main()
{
int i;
        for (i=100;i>=1;i--)
        {
        printf ("Hello BSD_Day 2011 in Bratislava!\n");
        printf ("This is RetroBSD on PIC32MX demonstration!\n");
        printf ("Countdown runs: %d\n\n", i);
        }
        return 0;
}
```

```
file hello.c line 11
```