Betriebssysteme

Entwicklungsmethodik

Literate Programming

Slide set 3:
Unix features

TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Winter semester 2015/16

Dr. Hans-Georg Eßer

h.g.esser@gmx.de

http://ohm.hgesser.de/

v1.2, 10/04/2015

Unix features: overview (1)

- Like all (modern) OSs, Unix offers an abstraction of the hardware and an interface through which processes can use it
- Processor (multi-tasking, scheduling)
- Process management (fork, exec, wait, exit)
- Hard disk / floppy disk (file system)
- Process communication and synchronization
- Network: TCP / IP, sockets
- Multi-user operation

Processor (scheduling)

- Unix makes the CPU resource available to several processes
- different scheduling strategies depending on the Unix version

- Prioritization of processes (nice values),
  Control over nice () and setpriority ()
- no process can monopolize the CPU

Process management (1)

- Unix processes organized in a tree structure ("father / son")
- New processes are included fork () as an identical copy of the calling process
- Process can with exec () reload another program
- exit () ends process (with return value)
- wait (), waitpid () wait for the end of a child process, evaluate the return value

Process management (2)

- Internally managed Unix processes via process control blocks (PCBs)
- each process has its own address space (Memory protection), which is also separated from the address space of the Unix kernel
- traditional: no threads (only via user-level library)
- Context switch changes from one process to the next (scheduler decides when and to which process)

File system (1)

- All Unix file systems have common properties

  - hierarchical file system (subfolders)
  - Media integrated in tree via mount points
  - Manage inodes Properties and block list of a file contain inodes ***none*** Filenames
  -
  - Directories are special files with filename mappings → Inode number

  - free data blocks and free inodes managed in bitmaps
  - Superblock contains management information for the entire file system

File system (2)

- System calls for access to files
  - fd = open () opens file, returns file descriptor
  - read (fd, ...) reads from file
  - write (fd, ...) writes to file
  - lseek (fd, ...) jumps to position in file
  - close (fd) closes file
- Internal implementation often via virtual file system (VFS)
- supports several FS and various hardware

IPC, process synchronization

- Signal mechanism
  - Signal handler, signal()
  - Send signal, kill ()
- BS itself also uses signals
- Synchronization via mutexes and / or semaphores

network

- TCP / IP (connections), UDP (connectionless)
- Sockets (client / server)
- Use of sockets similar to accessing files,
  - socket (),
    TCP: SOCK_STREAM,
    UDP: SOCK_DGRAM
  - I / O multiplexing with select ()

Multi-user operation

- Unix knows several users, root (uid = 0) is the system administrator
- Registration via username, password user
- groups
- File access: Access rights (for owner, group, other)
- Signal transmission: only to own processes

ULIX: Features, Limitations (1)

ULIX 0.13 is the first published release (Sept. 2015). In the course:

ULIX 0.08 and mini versions

- **Processes:**
  - fork, exit, wait, exec function;
    RR scheduler
  - current version without priorities (→ BA, done)
- **File system:**
  - VFS, hardware: floppy disks, IDE disks, Minix-v2-FS
  - mkdir, rmdir missing in version 0.08, symlinks
    only partially supported

ULIX: Features, Limitations (2)

- **Signals:**
  - Data structures in the PCB, signal()
    enters handler, kill () sends signal
  - received signals are ignored
    (except SIGKILL)
- **Storage:**
  - Paging: page tables for each process,
    Kernel address space becomes visible when changing to kernel mode
  - Outsourcing / re-storage of pages (from V. 0.12)

ULIX: Features, Limitations (3)

- **System calls:**
  - System call interface, allows registration
    new syscall handler.
    Standard syscalls for all kernel functions available

    User mode library contains functions
    which call these syscalls
- **Interrupts:**
  - Interrupt handler for timer, FDC, IDE, keyboard,
    serial interface

ULIX: Features, Limitations (4)

- **Access to devices:**
  - Floppy: DMA, block transfer
  - Hard disk: PIO, block transfer
  - "Serial Harddisk": byte-wise transfer via the serial interface
  - no device files (/ dev / *), available from 0.12
- **Network:**
  - not implemented (in BA: IP via SLIP)

ULIX: Features, Limitations (5)

- **Multi-user operation:**
  - Data structures available
  - effectively no multi-user operation possible, no rights check, no login (always root) (from v. 0.12: login, see below)
- **Terminals:**
  - ULIX launches shells on several terminals
  - no graphics mode; Terminals run in 80x24 text mode (line 25: BS status line)

Ulix 0.08 statistics

```
$ wc -lc ulix.c printf.c start.asm
      9155 283752 ulix.c                    // kernel
       185   4036 printf.c                  // printf-implem.
       355   9244 start.asm                 // boot, IRQs
      9695 297032 total
```