

# Digital UNIX

André Kloth, Sven Widmer

Hasso-Plattner-Institut  
Universität Potsdam

13. Juli 2006



# Agenda

- 1 Digital Equipment Corporation
- 2 Digital UNIX
- 3 Mach Interna
- 4 Präsentation
- 5 Zusammenfassung
- 6 Quellen

# Outline

- 1 Digital Equipment Corporation
- 2 Digital UNIX
- 3 Mach Interna
- 4 Präsentation
- 5 Zusammenfassung
- 6 Quellen

# Digital Equipment Corporation

- 1957 von Ken Olsen gegründet
- PDP Serie, noch heute im Einsatz
- VAX-Architektur, VMS und Ultrix
- erster kommerzielle verfügbarer 64-Bit RISC-Prozessor, Alpha
- Verkauf von Geschäftsteilen an Oracle, Intel und Cabletron
- 26. Januar 1998 durch Compaq übernommen



# Outline

- 1 Digital Equipment Corporation
- 2 Digital UNIX**
- 3 Mach Interna
- 4 Präsentation
- 5 Zusammenfassung
- 6 Quellen

## von der OSF bis zu HP

- *1988* - Beginn der Entwicklung von OSF/1 unter Leitung der OSF
  - DEC, IBM und Hewlett-Packard
  - Mach 2.5 Kernel der CMU mit BSD Kernel Service
- *1991* OSF/1 für MIPS-basierte DECstation
- *1992* Portierung auf Alpha AXP, OSF/1 AXP
- weitere auf OSF/1 basierte Systeme
  - AIX für System 370 und System 390
  - HP PA-RISC Workstation
- *1994* - OSF stellt Forschung und Entwicklung von OSF/1 ein

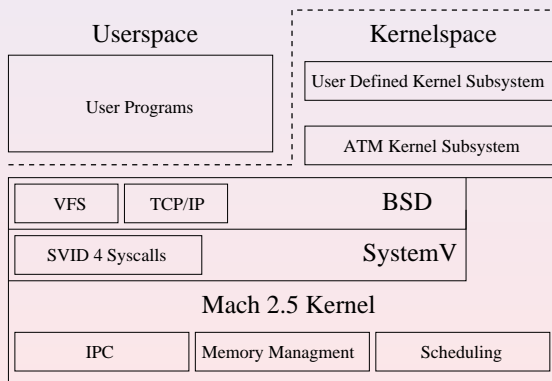
# von der OSF bis zu HP

- OSF/1 AXP wurde zu Digital Unix
- *1998* nach Übernahme von DEC durch Compaq Umbenennung in Tru64 UNIX
- ab *2002* unter HP Versuch Features von Tru64 in HP-UX zu übernehmen
- Support durch HP bis *2011*

# Digital UNIX

- Hybrid-Kernel
- Mach 2.5 Kernel
- 4.3BSD-Reno, 4.4 und SVID 4 kompatibel
  - TCP/IP, Sockets und VFS
  - Shared Libs, STREAMS Framework
  - /init.d
- Closed Source
- Stable Release: 5.1B-3 / 2. Juni 2005

# Architektur von Digital UNIX



# Outline

- 1 Digital Equipment Corporation
- 2 Digital UNIX
- 3 Mach Interna**
- 4 Präsentation
- 5 Zusammenfassung
- 6 Quellen

# Überblick

## Dienste im Kern

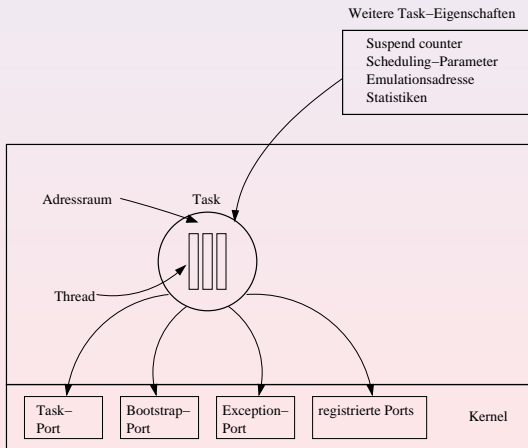
- Prozeßmanagement
- Speicherverwaltung
- Kommunikation
- I/O

## vorhandene Abstraktionen bzgl. User-space

- Tasks (Prozesse)
- Threads
- Speicher-Objekte
- Ports
- Nachrichten



# Prozeßmanagement



# Prozeßmanagement - *Tasks*

## einige Primitive

- `task_create`
- `task_terminate`
- `task_suspend`
- `task_resume`
- `task_assign`
- `task_assign_default`
- `task_info`
- `task_threads`
- `task_t` -> `priority`

# Prozeßmanagement - *Tasks*

## Unterschiede zwischen Mach Tasks und UNIX Prozessen

- UID, GID, Signalmask
- Root-Verzeichnis
- aktuelles Arbeitsverzeichnis
- Liste mit Datei-Handlern

# Prozeßmanagement - *Threads*

- heavyweight Threads
- *Thread Port* kann Threads im gleichen Task steuern
- User-space Interface C Threads
  - mappt nicht alle Funktionen der Kernel-Threads (— UNIX-Entsprechung)
  - `pthread_t pthread_fork(pthread_fn_t _func, void *_arg)` — `fork()`
  - `pthread_t pthread_exit(void *_result)` — `exit()`
  - `pthread_t pthread_join(pthread_t _thread)` — `waitpid()`
  - `pthread_t pthread_self()` — `getpid()`
  - `pthread_t pthread_detach(pthread_t thread)`
  - `pthread_t pthread_yield()` — `sched_yield()`

# Scheduling

- *Processor sets*
  - Sammlung von CPUs
  - eigene Ressourcen, "Benutzer" (Threads)
  - eigenes Scheduling
  - je PS 32 *run queues*

## kern/sched.h

```
#define NRQS 32      /* 32 run queues per cpu */

struct run_queue {
    queue_head_t    runq[NRQS]; /* one for each priority */
    decl_simple_lock_data(, lock) /* one lock for all queues */
    int    low; /* low queue value */
    int    count; /* count of threads runnable */
};

typedef struct run_queue *run_queue_t;
```

# Scheduling

- preemtiv
- Prioritätenbasiert 0-31
- globale Queue
- drei Prioritäten pro Thread

## kern/sched.h

```
struct thread {  
    ...  
    /* Scheduling information */  
    int    priority; /* thread's priority */  
    int    max_priority; /* maximum priority */  
    int    sched_pri; /* scheduled (computed) priority */  
    ...  
}
```

- `thread_handoff(...)` innerhalb des Kerns für Optimierungen



# Speicherverwaltung

- virtueller Speicher
- *memory object*
  - bezeichnet Seiten oder Dateien, etc.
  - können in den virtuellen Speicher gemappt werden
- drei Teile für Verwaltung notwendig
  - **pmap** — verwaltet MMU-Register, Frames und fängt alle Page faults ab (plattformabhängig)
  - plattformunabhängig; verschiebt Seiten, verarbeitet Seitenfehler (Algorithmen zur Seitenverwaltung)
  - **memory manager/external pager** — Verwaltung des virtuellen Speichers; welche Seiten befinden sich wo; Schnittstellen zum Kernel

# Speicherverwaltung - Memory Manager

- *object port*
- *control port*
- *name port*
  
- Memory Manager durch blockierendes `read()` des *control ports* suspended
- muss nicht lokal laufen
  
- copy-on-write bei *shared memory* Zugriffen von zwei Prozessen

# Ports und Messages

- Kommunikation zwischen Kern, Subsystemen, Tasks über Ports
  - les- oder schreibbar als *capability* (ähnlich zu UNIX Pipes)
  - Messages werden verschickt (Header, Body)
  - Threads innerhalb eines Tasks teilen sich ihre Ports
- *Network Message Server*
  - user-level Server (bei Mach 3.0 remote IPC und externe Speicherverwaltung im Kernel)
  - bildet *message ports* auf Netzwerk-Ports ab
  - kümmert sich um Datenkonvertierung, Authentifizierung, Lookup Service für Ports
  - Transparenz zwischen lokalen und entfernten Ports



# Outline

- 1 Digital Equipment Corporation
- 2 Digital UNIX
- 3 Mach Interna
- 4 Präsentation**
- 5 Zusammenfassung
- 6 Quellen

# Präsentation mit Digital UNIX 4.0

- Kernel-Modul
- Paketfilter



# Outline

- 1 Digital Equipment Corporation
- 2 Digital UNIX
- 3 Mach Interna
- 4 Präsentation
- 5 Zusammenfassung**
- 6 Quellen

# Zusammenfassung

- Was kennzeichnet Digital UNIX als modernes OS?
  - 64 Bit Kernel-Architektur (hybrid)
  - umfangreiche Kommunikationskonzepte
  - verteiltes Speichermanagement
  - ausgelegt für Multiprozessorsysteme

# FIN

- Danke für die Aufmerksamkeit!



# Outline

- 1 Digital Equipment Corporation
- 2 Digital UNIX
- 3 Mach Interna
- 4 Präsentation
- 5 Zusammenfassung
- 6 Quellen**

# Literaturverzeichnis



## The Mach Project Home Page

<http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/mach/public/www/mach.html>, Juli 2006.



## Distributed Operating Systems

Andrew S Tanenbaum, Prentice Hall, 1995.



## Tru64 UNIX – Online Documentation

<http://www.tru64.org/pages.php?page=Documentation>, Juli 2006.



## HP TestDrive

<http://www.testdrive.hp.com/>, Juli 2006.