BS I Übung 1

Aufgabe 1

* Verwaltung und Abstraktion der Hardware
  + Portabilität
* Verwaltung des Zugriffs auf Ressourcen
  + Sicherheitskonzept
  + Autorisierte Operationen auf autorisierte Daten
  + selektiv
  + Scheduling
  + User wollen nicht warten -> Geschwindigkeit
* Ausfallsicherheit (Bereitstellung einer zuverlässigen Umgebung für Anwendungen)
  + Absturz eines Benutzerprozesses sollte nicht zum Absturz des Systems/ anderer Prozesse führen
  + User wollen sich nicht mit Fehlern auseinandersetzen müssen
* Bereitstellung verschiedener Abstraktionsebenen (Subsysteme, Anwendungsebene, Systemebene, Hardwareebene)
  + Möglichmachen von high-level Programmierung
  + Erweiterbarkeit der Funktionen

1. Kernel-Mode:
   * CPU-, memory-, I/O-management (Treiber)
   * Multiprocessor management
   * Teile des Dateisystems und Netzwerkes
   * alles was den priviliged mode erfordert

User-Mode:

* + Fehleranfällige Prozesse
  + Benutzeranwendungen
  + Programmierumgebungen (inkl. Complier, Assembler, Interpreter, Linker/Loader)
  + Dateisystemverwaltung, Telekommunikation, Netzwerkverwaltung

1. Vorteile:
   * Behandlung von auftretenden Fehlern im Usermode
   * Zugriffsbeschränkungen für den Usermode
   * Bessere Abstraktionsmöglichkeiten
   * Sicher
     + Crash im Usermode gefährdet nicht das gesamte System

Nachteile:

* + Verwaltungsoverhead
    - -> Verzögerung bei Ausführung
  + Hohe Kosten bei Moduswechsel
  + Code im Kernelmode ist aufwändig und unflexibel

Aufgabe 2

|  |  |  |  |
| --- | --- | --- | --- |
| Programm | Prozess | Thread | Task |
| Folge von Anweisungen geschrieben in einer Programmiersprache | Instanz eines laufenden Programms | Prozess wird über Threads ausgeführt (Ausführungskontext eines Prozesses)   * Liegen auf selbem Speicher, Teilen Informationen | Menge von Programminstruktionen, die in Speicher geladen werden und abgearbeitet werden müssen |
| Ausführbare Programmdatei, die auf Datenträger liegt | Hat eigenen Adressraum, eigene Ressourcen, eigenes Sicherheitsprofil | Werden zur Ausführung geschedulet | Oberbegriff von Prozess und Thread |
|  |  | Besteht aus:  ID, Program Counter, Register Set, Strack |  |

Unterschiede:

Verwaltungsaufwand für Threads geringer als für Prozesse (teilen sich Informationen).

Aufgabe 3

* Ordner verwalten Funktionalitäten zu einem bestimmten Modul des BS
  + -> Module werden durch Verzeichnisse geordnet
* Struktur gibt keinen Überblick über Hierarchie
* Windows:
  + CONFIG = registry
  + DBGK = debugger service (user mode)
  + EX = handle management, pool allocator
  + FSRTL = Plug and Play Mgr.
  + IO = I/O-Mgr.
  + KE = Exception Handling, Scheduler, CPU Mgm
  + LPC = Local Procedure Call
  + MM = Memory Management
  + OB = Object Mgr.
  + PS = Processes and Threads (CREATE.c)
  + SE = Security Reference Monitor
* Linux:
  + Kernel enthält Prozesse, Scheduling, signals …
  + Kernel/gcov enthält gcc
  + Kernel/irq = Interrupts
  + Power = power mgr.
  + Mm = virtual memory mgm
  + Net = network protocols
* Windows-API DLLs( Kernel32, User32, Gdi32, etc…)
* Core System Processes (Smss, Winlogon, Services, Lsass)
* HAL
* Gerätetreiber
* Netzwerk-Stacks
* Linux: linux-3.2.64\kernel\sched.c
  + Sched.c
  + Sched\_autogroup.c
  + Sched\_clock.c
  + Sched\_debug.c
  + Sched\_cpupri.c
  + …
* Windows: base/ntos/ke/
  + Thredsup.c (routines for thread object)
  + Idsched.c (idle scheduling)

Aufgabe 4

Windows:

* Zu POSIX, OS/2 und DOS API-kompatibel
  + Teile des Windows-Kernels direkt von VMS übernommen
* Process Scheduler verwendet gleiche Prioritätsstufen, wie VMS
* Windows und VMS unterstützen Symmetric Multiprocessing (SMP)
* Chkdsk wird von DOS, OS/2 und Windows unterstützt

Linux:

* Übernimmt viele Funktionen aus der POSIX-API
* Benutzt den GNU-C-Compiler
* Linux orientiert sich an UNIX-Shell

Bzw beide:

* Time-Sharing (Mehrere User können gleichzeitig an einem Computer arbeiten)
* Virtuelle Speicherverwaltung:
  + Jeder Prozess bekommt eigenen virutellen zusammenhängenden Speicherbereis
  + -> Prozesse können nicht versehentlich auf fremden Speicher zugreifen
* Kommandozeileninterpreter:
  + User kann über eine Schnittstelle Programme oder Funktionen aufrufen
  + Windows und Linux stellen Shell zur Verügung