

Aufgabe 1: Verständnisfragen

1. Im Kernmodus wird mit unbeschränkten Berechtigungen ausgeführt: Man kann jede Funktion aufrufen, auf jede Ressource zugreifen und nutzt Kernthreads. Im Benutzermodus hat man nur beschränkte Rechte und alle Aktionen die solche erfordern werden über den Kernmodus via System Call Interface ausgeführt.
2. Welche Befehle sollten nur im Kernmodus ausgeführt werden?
 - Nur Kernmodus, da der Hardware zugriff dadurch geregelt wird
 - Auch Benutzermodus, die Systemzeit zu lesen beeinflusst keinen anderen Prozess oder das Betriebssystem
 - Nur Kernmodus, da die Systemzeit viele andere Prozesse beeinflusst
 - Nur Kernmodus, da hier Code und Daten anderer Prozesse direkt beeinflusst werden
3. Vorteile: Konsekutive logische Adressierung, möglich mehr Arbeitsspeicher zu Verfügung zu stellen als vorhanden, mehrere Programme können gleichzeitig ausführen mit geringem Mehraufwand. Nachteile: Mehraufwand beim Laden, komplexere Speicherverwaltung, z.T. kein bulk laden möglich
4. Lesen von der Datei bevor zuende geschrieben wurde führt zu einem invaliden read. Der Treiber könnte dieses Problem durch locks lösen bspw. durch Read-Write Locks.

Aufgabe 2: Selbststudium

1. Charaktersitisch für Betriebssystem-Art:
 - Mehrere verschiedene Anwendung mit Nutzeroberfläche müssen gleichzeitig ausgeführt werden & die Nutzeroberfläche muss immer "Responsive" sein, d.h. in kurzmöglichster Zeit reagieren. Meist nur ein oder wenige Nutzer die physisch vor dem Gerät sitzen. Häufige Änderung der Konfiguration (wie Programme installieren, Downloads,...)
 - Oft real-zeit kritische Anwendungen meist ohne Nutzeroberfläche müssen sehr robust und zuverlässig Funktionieren. Wird oft in Maschinen eingesetzt, wie z.B. Autos oder Robotern. Sollte nach einmaligem Setup nicht mehr regelmässig geändert werden müssen.
 - Sendet Daten über das Netzwerk an einen Empfänger oder speichert große Mengen der gemessenen Werte. Führt meist nur einen Prozess aus pro Sensor und keine weiteren Funktionen. Zugriff haben meist nur wenige Zugriffsberechtigte.
 - Wird oft in Kombination mit eingebetteten Systemen genutzt. Immer dann wenn es essenziell ist, dass das System reagiert wie z.B. beim Bremsen eines Autos

- Ein Time-Sharing System führt die Prozesse eines Nutzers eine festgelegte Zeit aus und bedient dann den nächsten Nutzer. Ein Multiprogrammsystem führt einen Prozess eine festgelegte Zeit aus und bedient dann den nächsten Prozess unabhängig vom Nutzer.

Aufgabe 3: Einheiten

1.	yobi	Yi	80	8
	zebi	Zi	70	7
	exbi	Ei	60	6
	pebi	Pi	50	5
	tebi	Ti	40	4
	gibi	Gi	30	3
	mebi	Mi	20	2
	kibi	Ki	10	1

- Weil halbe Bits physisch nicht repräsentierbar sind. Komma Zahlen werden anders codiert. Üblicherweise Größenangaben für Speicher.
- 0,909495

Aufgabe 4: Ausführungs- und Zugriffszeiten

- 333333 Befehle pro Sekunde
 - 200000 Befehle pro Sekunde
- $0.95 \cdot 2ns + 0.05 \cdot (0.99 \cdot 10ns + 0.01 \cdot 10000ns) = 7.395ns$
 - Die Zugriffszeit zum externen Speicher, da sie um das 5000-fache größer ist als die Cache Zugriffszeit bzw. das 1000-fache als die Arbeitsspeicherzugriffszeit. Ausserdem sind 95% Cache-Hit rate sehr optimistisch.
 - $0.95 \cdot 2ns + 0.05 \cdot (0.99 \cdot 10ns + 0.01 \cdot 2002,395ns) = 3.3961975ns$

Aufgabe 5: Hello World!

- Siehe hello.c
- `-std=c11` setzt den C standard bzw. die verfügbaren Sprachkonstrukte auf den Iso standard von 2011
 - `-g` generiert Debugging Informationen im nativen Format des Betriebssystems z.B. DWARF für Unixoiden
 - `-Wall` aktiviert die meisten Warnungen des Compilers
 - `-pedantic` aktiviert weitere Warnungen des Compilers
- `.bc` — Bitstream file, containing an abstract encoding of the intermediate representation in binary form
- `.i` — a preprocessed file: Macros are expanded and preprocessor directives are executed.

5. `.s` — An assembly file, architecture specific instructions
6. `.o` — A machine code file, binary format, specific to the architecture, generated by execution of the assembler with the `.s` file as input
7. `.out` — linked machine code file, ready for execution, gained by execution of the linker with the `.o` file as input

Aufgabe 6

1. It tells the linker to link against the `c math` library
2. store the image in a two dimensional array and print it.