

## 1 Thema: Systemmodus (6 Punkte)

### 1.1 Systemmodus

Es wurde gesagt, dass Unix für Echtzeitanwendungen ungeeignet sei, weil ein Prozess, der im Systemmodus läuft, nicht verdrängt werden kann.

1. Beschreiben Sie in einigen Sätzen, wodurch sich Echtzeitanwendungen vom interaktiven Benutzen eines Rechners unterscheiden.
2. Nennen Sie zwei typische Anwendungsklassen.
3. Welche Eigenschaft muss die Prozessverwaltung hier zusätzlich aufweisen?

## 2 Thema: Threads (12 Punkte)

### 2.1 Konzepte (4 Punkte)

Erklären Sie die allgemeinen Konzepte von Prozessen und Threads wie sie in Betriebssystemen wie z.B. Linux oder Windows realisiert sind [Schreiben Sie maximal eine halbe Seite]

### 2.2 POSIX-Threads (8 Punkte)

Mit Hilfe von POSIX-Threads wurde das unten abgedruckte Programm realisiert. Es sollte sich auf einem Linux-Rechner folgendermaßen übersetzen lassen:

```
gcc -o <NameOfBinary> <NameOfCFile.c> -lpthread.
```

Das Programm startet drei Threads mit jeweils unterschiedlichen Namen. Jeder Thread gibt seinen Namen aus und den aktuellen Wert der Variablen `value`.

Wir betrachten nun die drei Ausgabezeilen, die dieses Programm auf dem Bildschirm erzeugt.

- In welcher Reihenfolge erscheinen die drei Farbnamen in der Ausgabe? Gibt es mehr als eine Reihenfolge? Wenn ja, wieviele? Begründen Sie Ihre Aussage.
- Wenn wir die Sequenz der nacheinander ausgedruckten Variablenwerte betrachten, wieviele verschiedene Wertefolgen gibt es dann hier? Begründen Sie Ihre Aussage.
- Beachten Sie, daß auf einem Rechner mit nur einem Prozessor, die für Threads typischen Effekte möglicherweise nicht beobachtbar sind. Das Programm läuft natürlich korrekt. Erläutern Sie kurz, welche Effekte wir erwarten würden und warum sie unter Umständen nicht auftreten.

```

#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>

void saying_thread_name( void *ptr );
int value=1;

int main(void)
{
    pthread_t wthread, bthread, rthread;
    char *wname = "white";
    char *bname = "black";
    char *rname = "red";
    pthread_attr_t *pthread_attr_default = NULL;

    pthread_create( &wthread, pthread_attr_default,
        (void*)&saying_thread_name, (void*) wname);
    pthread_create(&bthread, pthread_attr_default,
        (void*)&saying_thread_name, (void*) bname);
    pthread_create(&rthread, pthread_attr_default,
        (void*)&saying_thread_name, (void*) rname);

    pthread_join(wthread,NULL);
    pthread_join(rthread,NULL);
    pthread_join(bthread,NULL);
    exit(0);
}

void saying_thread_name( void *ptr )
{
    char *meldung;
    meldung = (char *) ptr;
    printf("I am a %s thread and the value is %d \n", meldung, value);
    value=value+1;
}

```

### 3 Thema: Kernel-Architekturen (12 Punkte)

Beim Aufbau von Betriebssystemen unterscheidet man die Kernelarchitekturen Monolithischer Kernel, Minimaler Kern (Mikrokern) und Hybridkernel (Makrokern). Beschreiben Sie die Merkmale dieser drei Kernelarchitekturen. Worin unterscheiden sich diese Kernelarchitekturen und was sind die Vor- und Nachteile der unterschiedlichen Kernelarchitekturen? Zeichnen Sie zu jeder der drei Kernelarchitekturen ein Diagramm. Nennen Sie für jede der drei Kernelarchitekturen mindestens ein Betriebssystem, das diese Architekturen verwendet.