



Betriebssysteme: Konzepte, Dienste,
Schnittstellen
(Betriebssysteme und betriebssystemnahe
Programmierung)

SS 2005 – Übungsblatt 7

Ausgabe: 6. Juni 2005

Abgabe: bis spätestens 13. Juni 2005
im Fachschaftsraum Mathematik
oder per email an c.markmann@uni-wuppertal.de

Aufgabe 1. *stty und system()*

Übersetzen Sie das folgende kleine Programm

```
////////////////////////////////////  
// Datei:   inchar.cc  
// Version: 1.0  
// Autor:   Hans-Juergen Buhl  
// Datum:   3. 6. 2003  
////////////////////////////////////  
  
#include      <iostream>  
#include      <string>  
#include      <cassert>  
  
#include      <cstdlib>  
  
using namespace std;  
  
int main()  
{
```

```

char inp('\000');

system("clear");
cout << endl << endl << "Please press any key: ";

system("stty raw -echo");
cin >> inp;
system("stty sane echo");

cout << endl << endl
    << " Die Taste mit dem Code " << static_cast<int>(inp)
    << " wurde betaetigt" << endl << endl;
cout << "Ende\n" << endl << endl;

}

```

und informieren Sie sich mittels `man` über die benutzten Optionen des Kommandos `stty`. Beschreiben Sie dann die Wirkungsweise des Programms. Was geschieht, wenn Sie den Aufruf von `system("stty sane echo")` vergessen?

Welche Funktion kommerzieller Software kann man auf diese Art verwirklichen? Schreiben Sie ein kleines Beispielprogramm, das diese Funktion demonstriert.

Aufgabe 2. *isatty*

Bringen Sie das folgende Programm zum Ablauf

```

#include <termios.h>
#include <unistd.h>

int
main(void)
{
    struct termios term;
    long vdisable;

    if (isatty(STDIN_FILENO) == 0){
        perror("standard input is not a terminal device");
        exit(1);
    }

    if ( (vdisable = fpathconf(STDIN_FILENO, _PC_VDISABLE)) < 0){
        perror("fpathconf error or _POSIX_VDISABLE not in effect");
        exit(1);
    }

    if (tcgetattr(STDIN_FILENO, &term) < 0){ /* fetch tty state */
        perror("tcgetattr error");
    }
}

```

```

        exit(1);
    }

    term.c_cc[VINTR] = vdisable;    /* disable INTR character */
    term.c_cc[VEOF]  = 2;          /* EOF is Control-B */

    if (tcsetattr(STDIN_FILENO, TCSAFLUSH, &term) < 0){
        perror("tcsetattr error");
        exit(1);
    }

    exit(0);
}

```

und erklären Sie es Zeile für Zeile. Überprüfen Sie dessen Wirkungsweise mit Hilfe des Kommandos `stty -a`. Wie kann man den gleichen Effekt interaktiv mit Hilfe von `stty` erzeugen?

Überlegen (und nennen) Sie auch hier Anwendungsmöglichkeiten in Anwenderprogrammen!

Aufgabe 3. *winsize*

Bringen Sie das folgende Programm zum Ablauf

```

#include      <signal.h>
#include      <termios.h>
#ifdef TIOCGWINSZ
#include      <sys/ioctl.h>    /* 44BSD requires this too */
#endif
#include      <unistd.h>

static void   pr_winsize(int), sig_winch(int);

int
main(void)
{
    if (isatty(STDIN_FILENO) == 0)
        exit(1);

    if (signal(SIGWINCH, sig_winch) == SIG_ERR){
        perror("signal error");
    }

    pr_winsize(STDIN_FILENO);    /* print initial size */
    for ( ; ; )                 /* and sleep forever */
        pause();
}

static void

```

```

pr_winsize(int fd)
{
    struct winsize  size;

    if (ioctl(fd, TIOCGWINSZ, (char *) &size) < 0){
        perror("signal error");
    }

    printf("%d rows, %d columns\n", size.ws_row, size.ws_col);
}

static void
sig_winch(int signo)
{
    printf("SIGWINCH received\n");
    pr_winsize(STDIN_FILENO);
    return;
}

```

und erklären Sie seine Wirkungsweise Zeile für Zeile. Was ist hier das Haupteinsatzgebiet?

Aufgabe 4. *SystemAndDate*

Arbeiten Sie analog mit:

```

////////////////////////////////////
// Datei:   SystemAndDate.cc
// Autor:   Hans-Juergen Buhl
// Datum:   3. 6. 2003
//
// Bemerkung: evtl. explizit mit -lnsl uebersetzen
//
////////////////////////////////////

#include      <iostream>
#include      <string>

#include      <time.h>           // oder ctime
#include      <locale.h>        // oder clocale

#include      <stdio.h>          // oder cstdio

#include      <unistd.h>
#include      <netdb.h>

using namespace std;

class Nachricht {

```

```

const string Text;

string CreateTime;
string Hostname;
string Domainname;
string Username;

public:

    Nachricht(const string& t);

    void print() const { cout << Text; };

    void printSignature() const { cout << "Signatur: " << CreateTime
        << "          " << Username << "@"
        << Hostname << "." << Domainname; };

};

Nachricht::Nachricht(const string& t) : Text(t), Username(cuserid(NULL))
{
    time_t Now = time(NULL);
    setlocale(LC_ALL, "de");
    CreateTime = ctime(&Now);

    char host[20];
    gethostname(host, sizeof(host));
    Hostname = host;

    // char domain[40];
    // getdomainname(domain, sizeof(domain)); //funktioniert nicht!warum?
    {
        struct hostent *ho;
        ho = gethostbyname(host);
        // clog << ho->h_name << endl;
        char *domain = strchr(ho->h_name, '.');
        if (domain != 0) domain++;
        // clog << (long)domain << endl;

        if (domain==0)
            Domainname="";
        else
            Domainname = domain;
    }
}

int main()
{
    Nachricht N1("... Meine erste automatisch signierte Nachricht ...");

```

```
N1.print();
cout << endl << endl;

N1.printSignature();
cout << endl;

return 0;
}
```

Ergänzen Sie dann überall dort, wo nötig, die Abfrage des Erfolgszustandes der aufgerufenen Funktionen. Warum ist das erforderlich?

Welche Einsatzgebiete sehen Sie für die hier demonstrierte Funktionalität?

Aufgabe 5. *virtuelle Betriebssysteme*

Erläutern Sie die in der Vorlesung diskutierten verschiedenen Typen virtueller Betriebssysteme und stellen Sie in einer Tabelle zusammen, für welche Aufgaben jeder Typ besonders und für welche weniger geeignet ist.

Welche anderen virtuellen Maschinen aus

http://en.wikipedia.org/wiki/Comparison_of_virtual_machines

erscheinen Ihnen darüber hinaus noch interessant? Warum?