

# Vorlesung Betriebssysteme

Prof. Dr. Jan Dünneberger

Verteilte Systeme und Betriebssysteme  
Fakultät für Informatik und Mathematik  
Ostbayerische Technische Hochschule Regensburg

Wintersemester 2013/14

# Valgrind - Speicherfehler aufdecken

- Folgendes C-Programm enthält 2 Fehler:

## Beispiel für dynamische Speicherverwaltung

```
#include <stdlib.h>

void f(void) {
    int *x = malloc(10 * sizeof(int));
    x[10] = 0; }

int main(int argc, char **argv) {
    f();
    return 0; }
```

- Sehen Sie sich zunächst einen *Backtrace* zu dem Laufzeitfehler im gdb an und beheben Sie diesen
- Erstellen Sie ein Profil mit `valgrind --leak-check=yes ./main` und beseitigen Sie das Speicherleck
- Machen Sie sich auf [valgrind.org/docs](http://valgrind.org/docs) näher mit Valgrind vertraut und überprüfen Sie Programme aus früheren Übungen

# Helgrind - Thread-Probleme beheben

- Folgendes C-Programm enthält eine Race Condition

## Data Race

```
#include <pthread.h>

int var = 0;
void* child_fn ( void* arg ) {
    var++; return NULL; }
int main ( void ) {
    pthread_t child;
    pthread_create(&child, NULL, child_fn, NULL);
    var++; pthread_join(child, NULL);
    return 0; }
```

- Erstellen Sie ein Profil mit `valgrind --tool=helgrind` und analysieren Sie die Ausgabe
- Machen Sie sich auf [valgrind.org/docs/manual/hg-manual.html](http://valgrind.org/docs/manual/hg-manual.html) mit Helgrind vertraut und überprüfen Sie die Beispiele aus der letzten Übung (`bug1.c` und `bug2.c`) mit den Pthread-Conditions