

Übungsunterlagen zur Übung
Softwareentwicklungswerkzeuge
Übungseinheit – aiT

Inhaltsverzeichnis

1.	Zielstellung der Übung	3
2.	Die Aufgabenstellungen	4
2.1.	1. Aufgabe	4
2.2.	2. Aufgabe	4
3.	Hinweise zur Arbeit mit aiT	6
3.1.	Kompilierung mittels GCC.....	6
3.2.	Arbeit mittels aiT	8
3.2.1.	Laden der Executables.....	8
3.2.2.	Durchführen der Analyse	10
3.2.3.	Nähere Spezifikationen mittels .ais – Dateien.....	11

1. Zielstellung der Übung

Ziel der Übung ist die Auseinandersetzung mit dem Tool aiT mit dessen Möglichkeiten, die maximale Laufzeitschranke von vorhandenen Binaries automatisch berechnen zu lassen.

Dabei soll jeder Student kleine C-Programme entwickeln und diese anschließend mittels GCC kompilieren. Die entstehenden Binaries sollen mittels aiT untersucht und die Worst Case Execution Time automatisch berechnet und angezeigt werden. Dabei sind die verschiedenen Darstellungsmöglichkeiten (Pipelineverhalten, Cacheverhalten) mit zu berücksichtigen und anzusehen. Bei der Variante mit den Schleifen muss außerdem noch die maximale Schleifendauer eingegeben werden, da aiT diese bei der in der Übung verwendeten Art nicht berücksichtigen kann.

Wenn am Schluss noch Zeit ist, sollen die Studenten Veränderungen am Code vornehmen, und die Auswirkungen auf die WCET ermitteln, z.B. einen extra Funktionsaufruf einbauen, mehr Variablen deklarieren, etc. .

2. Die Aufgabenstellungen

Die Aufgabenstellungen beinhalten 2 kleine C-Programme, die mittels des windowseigenen Editors oder dem Editor Proton entwickelt und mit GCC kompiliert werden sollen. Anschließend sollen die entstehenden Binaries mit aiT ausgewertet werden.

2.1. 1. Aufgabe

- (a)** Für einen Geldausgabeautomaten ist ein Programm in C mit folgenden Eigenschaften zu entwickeln:
- i. Als Eingabe wird eine natürliche Zahl erwartet
 - ii. Dieser Eingabebetrag n soll in Vielfachheiten gültiger Geldbeträge umgerechnet und diese ausgegeben werden (gültige Beträge: 500, 200, 100, 50, 10, 5, 2, 1)
- (b)** Ermitteln Sie mittels aiT die WCET Ihrer Programmvariante. Nutzen Sie dabei auch die verschiedenen grafischen Darstellungen des Tools. Erstellen Sie dabei eine .ais – Datei, mit deren Hilfe Sie die Prozessormerkmale und die Programmmerkmale genauer spezifizieren.

2.2. 2. Aufgabe

- (a)** Schreiben Sie in C ein Programm, welches die n -te Fakultät berechnet. Kompilieren Sie anschließend dieses Programm mittels gcc. Realisieren Sie dabei 2 Varianten: mittels Funktionsaufruf und mit Realisierung in der main-Routine.

Alternativ:

Entwickeln Sie ein Programm in C, was die k-te Fibonacci-Zahl berechnen kann. Schreiben Sie dafür eine rekursive und eine iterative Lösungsvariante und kompilieren Sie anschließend diese in gcc.

Hinweis:

Funktionsdefinition $fib(n)$:

$$fib(0) = 0$$

$$fib(1) = 1$$

$$fib(n) = fib(n-1) + fib(n-2)$$

- (b) Ermitteln Sie mittels aiT die WCET der beiden Varianten und vergleichen Sie beide. Nutzen Sie dabei auch die verschiedenen grafischen Darstellungen des Tools. Erstellen Sie dabei eine .ais – Datei, mit deren Hilfe Sie die maximalen Schleifendurchläufe und die maximale Tiefe der Rekursion einstellen können und experimentieren Sie mit verschiedenen Werten. Was passiert ohne eine .ais-Datei?

Bitte beachten Sie zum Lösen der Aufgaben die Hinweise im nachfolgenden Abschnitt.

3. Hinweise zur Arbeit mit aiT

3.1. Kompilierung mittels GCC

Die Zielplattform des hier vorliegenden aiT ist der Motorola PowerPC. Um entsprechende Binaries (ausführbare Dateien) für diese Plattform zu erhalten, ist eine Kompilierung mit einem Crosscompiler unter gcc notwendig.

Als Plattform, unter der dieser Crosscompiler abläuft, dient die Linuxumgebung des Rechenzentrums.

Folgende Vorgehensweise ist notwendig:

- (1) Kopieren der erstellten .c –Quelldatei in ihren Home-Ordner oder einem Unterordner Ihres Verzeichnisses
- (2) Starten Sie Putty, um eine SSH-Verbindung zum Linuxserver „ganymed“ herzustellen. Dabei einfach im aufgehenden Fenster Doppelklick auf „ganymed“
- (3) Wechseln Sie in das Verzeichnis Ihres Homeordners, unter dem Sie die .c – Quelldatei gesichert haben (Befehlseingabe mittels cd <Verzeichnis>, um unterliegendes Verzeichnis zu öffnen bzw. mit cd .. Wurzelordner öffnen)
- (4a) Um den erstellten Code zu testen, ob dieser die Anforderungen erfüllt, kann mittels gcc eine ausführbare Datei erstellt werden:

gcc <Ihre C-Datei - Name>.c –o <Ausgabedateiname>

Hierbei können einfache printf und scanf – Befehle zum Anzeigen bzw. Einlesen von Werten benutzt werden.

Ausführen der Datei erfolgt mittels

./<Ausgabedateiname>

- (4b) Um eine ausführbare Datei für den Power PC zu generieren, die mittels aiT weiterverarbeitet werden kann, geben Sie folgende Befehlsfolge ein (nach erfolgreichen Testen des Codes):

**/opt/crosstool/gcc-4.1.0-glibc-2.3.6/powerpc-750-linux-gnu/bin/powerpc-750-
linux-gnu-gcc <Ihre C-Datei - Name>.c -g -o <Ihre ausführbare Datei - Name>.elf**

Bitte beachten Sie jetzt, das jegliche printf und scanf – Routinen entfernt werden müssen, da hierbei aiT mit der WCET-Analyse Probleme hinsichtlich der Auswertbarkeit bekommt.

Jetzt wird im selben Verzeichnis die <Dateiname>.elf Datei erstellt, mit der Sie in aiT weiterarbeiten können.

3.2. Arbeit mittels aiT

3.2.1. Laden der Executables

Nach dem Start von aiT erscheint folgendes Benutzermenü:

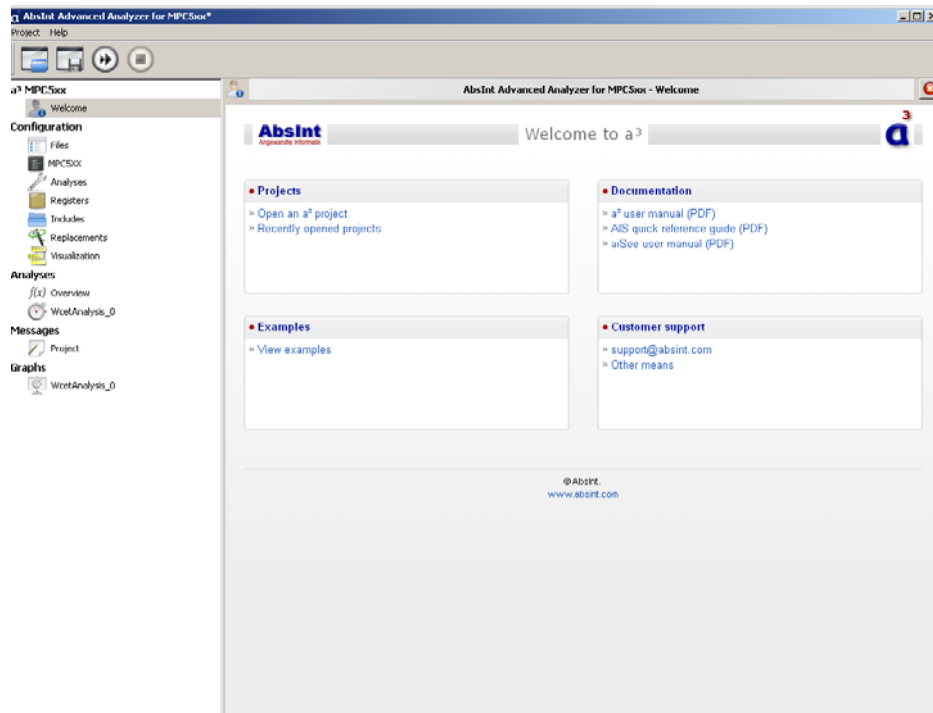


Abbildung 3.1

Hier können Beispiele und die Dokumentation angesehen werden. Auf der linken Seite finden sich alle wichtigen Einstellungsmöglichkeiten. Da der Compiler momentan die Datei ab Adresse 0x10000 linkt, ist eine Modifikation unter MPC5XX notwendig. In Abbildung 3.2. muss unter dem Registerreiter „Memory Controller“ die Option „valid“ gesetzt werden.

Weiterhin ist zu beachten, um den Sourcecode unter aiT im Graphen sehen zu können, müssen die kompilierte .elf-Datei sowie die Sourcedatei .c sich im selben Arbeitsverzeichnis befinden.

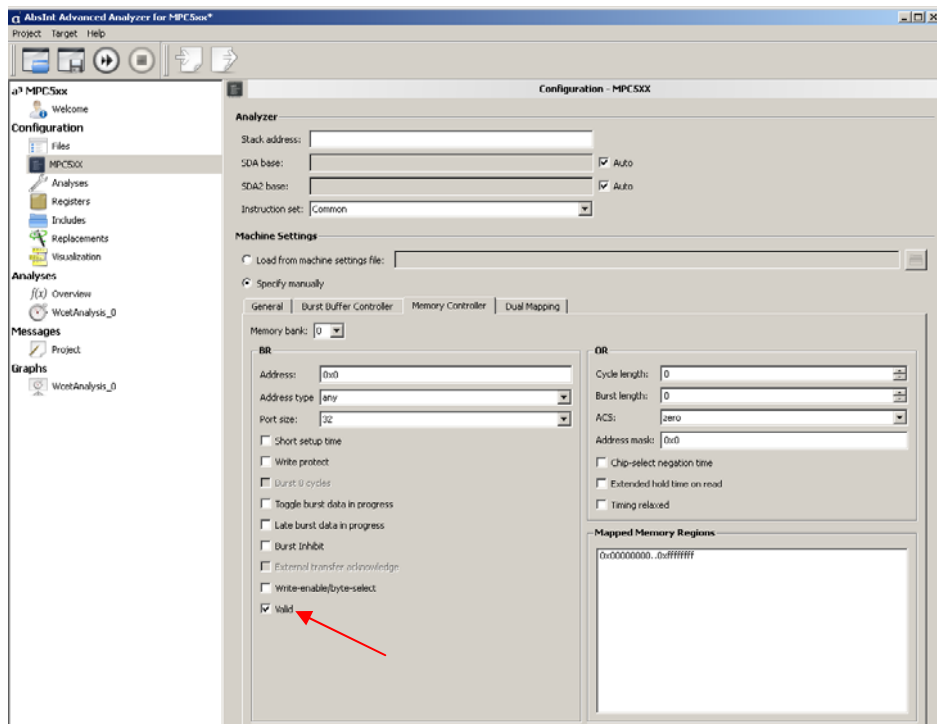


Abbildung 3.2

Jetzt kann wie in der nächsten Abbildung eine ausführbare Datei ausgewählt und die evtl. notwendige .ais – Datei hinzugefügt werden (Erläuterung der .ais- Files im nächsten Kapitel).

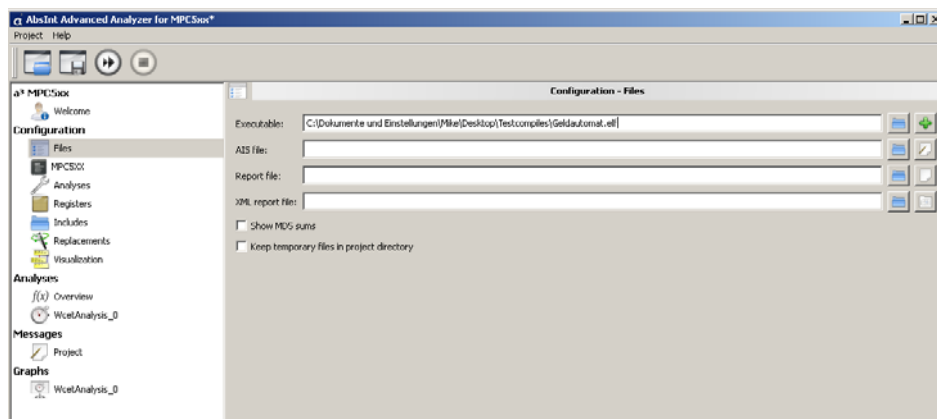


Abbildung 3.3

3.2.2. Durchführen der Analyse

Um eine WCET-Analyse durchführen zu können, müssen folgende Schritte durchgeführt werden:

- (1) Unter „Analyses“ und „Overview“ die WCET-Analyse auswählen
- (2) Auswahl des Einsprungpunktes (im Normalfall die Mainroutine):

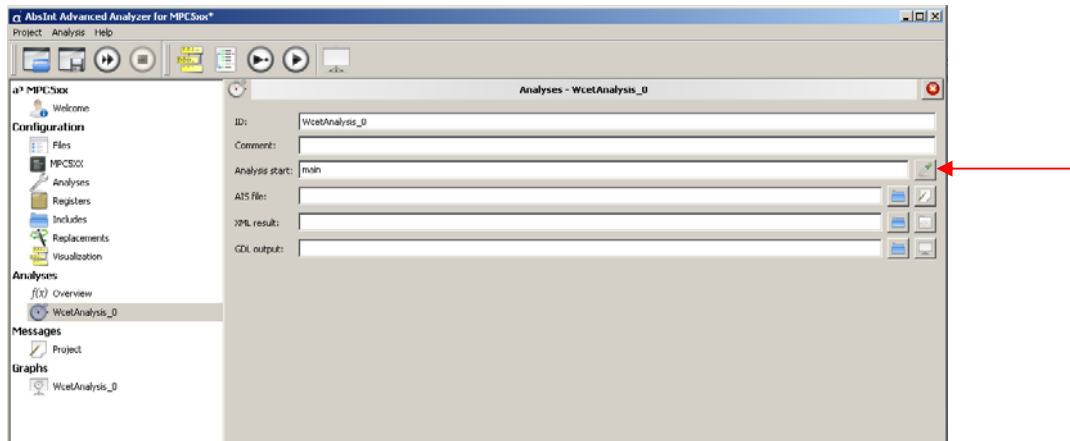


Abbildung 3.4

- (3) Durchführen der WCET-Analyse:

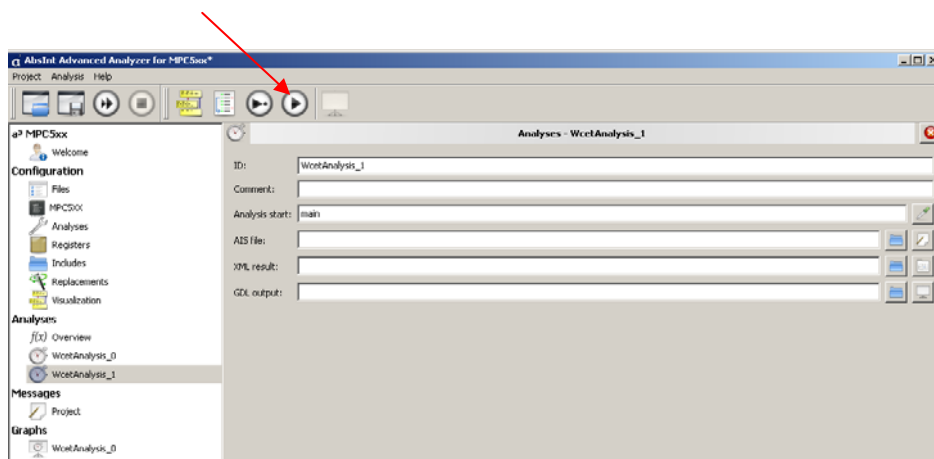


Abbildung 3.5

Ergebnis ist das folgende Fenster:

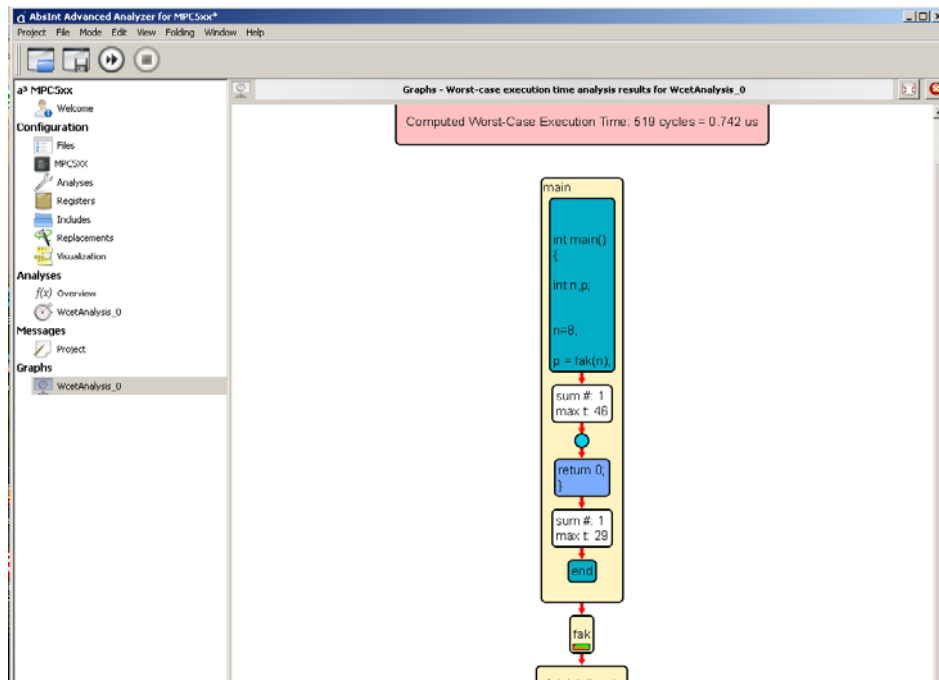


Abbildung 3.6

Hierbei wird die WCET-Time in Maschinenzyklen angegeben und die Zusammenhänge zwischen optionalen Methoden innerhalb des Quellcodes.

3.2.3. Nähere Spezifikationen mittels .ais – Dateien

Um nähere Angaben zu den verwendeten Prozessortypen und bei undefinierten Rekursionen und Schleifenaufrufen eine maximale Zeit festlegen zu können, ist es empfehlenswert, eine solche Datei ebenfalls im Editor anzulegen.

Wichtige Grundregeln:

- Terminiert wird jede Spezifikation mit einem Semikolon
- Befehle können sich über mehrere Zeilen erstrecken, einzelne Symbole werden mit Leerzeichen bzw. Leerzeile getrennt
- # dient zur Auskommentierung, Kommentar geht bis Zeilenende

Beispieldatei:

```
##
## Globale Einstellungen
```

```
##  
  
# compiler specification  
compiler "ppc-gcc";  
  
# Taktrate  
clock exactly 450 Mhz;  
  
# max Tiefe der Rekursion  
recursion "fib" max 9;  
  
# max Tiefe der Rekursion für Stack-Analyse  
routine "fib" incarnates max 4;  
  
Beispiel:  
# Schleifendurchläufe  
loop "main" + 1 loop max 5 end;
```

Weitere Spezifikationsmöglichkeiten und Anwendungen sind im Manual unter dem Punkt **5 AIS Annotations** zu finden.

Hierbei kann man bestimmte Einstellungen verschiedentlich ausprobieren und die Auswirkungen auf die WCET-Analyse sehen. Weitere Beispiele für .ais-Dateien sind im aiT-Verzeichnis unter Beispiele zu finden.