

1. Praktikum

1. Programmiertechnik und Softwareentwicklung

Das nachfolgend vorgegebene Assembler-Programm EUKLID03.asm ist in mehreren Schritten fortzuentwickeln und soll dabei aber in seiner Funktion unverändert bleiben.

1.Aufgabe

Es ist das folgende Assemblerprogramm EUKLIDxx.asm vorgegeben. In diesem Programm wird nach der Eingabe zweier vorzeichenlosen Dezimalzahlen < 65536 (16 Bit) mit Hilfe des **Euklid-Algorithmus** (siehe PAP im Anhang) der größte gemeinsame Teiler dieser beiden ganzen Zahlen ermittelt und das Ergebnis als Dezimalzahl auf den Bildschirm ausgegeben.

- Es ist im Hauptprogramm der fehlende Programm-Code des Euklid-Algorithmus-Kerns anhand des PAP im Anhang zu entwickeln und nachzutragen.
- Für das Unterprogramm zur dezimalen Zahlen-Ausgabe (Aus5Dez) soll der verwendete Algorithmus in Form eines PAPs und einer kurzen Erläuterung des zugrundeliegenden Schemas dargestellt werden. (siehe Rechnergrundlagen – Zahlenumwandlung)
- Es handelt sich bei dem vorgegebenen Programm um ein **.COM-Programm**, bestehend aus einem einzigen Segment für Programm und Daten sowie **vollständigen Segmentzuweisungen**. Dieses vorgegebene .COM-Programm mit seinen vollständigen Segmentzuweisungen ist zu einem .COM-Programm mit den **vereinfachten Segmentzuweisungen des Turbo-Assemblers (MODEL TINY)** umzugestalten.

Skript: Software Seite 9 – 15, Systemunterlagen Seite 4 – 6, Befehlssatz

2.Aufgabe

Bei gleichbleibender Funktion des EUKLIDxx sollen die darin verwendeten **DOS-Konsolfunktionen 01 (Eingabe) und 02 (Ausgabe) und 09 (Stringausgabe)** des INT 21h durch die entsprechenden BIOS-Funktionen für Tastatur INT 16h und Bildschirm INT 10h ersetzt werden.

(Fktn 13h des INT 10h hier sehr unkomfortabel !!!-warum???)

Dabei soll die mehrfach auftretende Stringausgabefunktion 09h INT 21h durch einen **Makro stringout**, das eine BIOS-Funktion verwendet, ersetzt werden. **Skript: Software Seite 7**

Nur die DOS-Beendigungs-Funktion 4Ch INT 21h kann beibehalten werden.

Hinweis: durch geschickte Wahl der BIOS-Funktion, kann für der Bildschirmausgabe eine einfache Lösung gefunden werden !

Skript zu DOS/BIOS-Fktn: Systemunterlagen Seite 12 – 30 und ausführlich in System-Funktionen

3.Aufgabe

Die gesamte Ein-Ausgabe soll jetzt ausschließlich mit Hilfe geeigneter Unterprogramme aus der gegebenen **UP-Bibliothek eados** (Ein- Ausgabe mittels DOS-Funktionen) durchgeführt werden.

Dazu muß eados.asm übersetzt und das Objekt-File eados.obj zum Hauptprogramm hinzugelinkt werden (TLINK ***** + eados).

Damit bei der Übersetzung des Hauptprogramms die auswärtigen UP-Bezeichner aufgelöst werden können, sollten die dafür nötigen EXTRN-Deklarationen in einer zusammenfassenden INCLUDE-Datei eados.inc am Beginn des Hauptprogrammes eingefügt werden

Hinweis: Zu beachten ist die unterschiedliche String-Terminierung der DOS-Fktn 09 und der entsprechenden UP-Bibliotheks-Routinen

Auch die beiden UP **aus16dez** und **ein5dez** sind durch eados-Bibliotheks-Routinen zu ersetzen!!

Dies ergibt dann natürlich die kürzeste EUKLIDxx-Quelldatei.

Skript: Software Seite 8-9

4. Aufgabe

Das vorgegebene 16-Bit-Programm EUKLIDxx verarbeitet mit seinen 16Bit-Operanden Zahlenwerte bis zu einer maximalen Größe von 65535

Es ist zu einem Programm abzuändern das **32Bit-Operanden** verarbeiten kann, d.h. der Zahlenbereich soll auf $4\,294\,967\,295 = (2^{32}) - 1$ erweitert werden.

Dazu müssen jetzt für die Berechnungen die 32-Bit-Register (EAX, EBX...) herangezogen werden.

Zu beachten ist hierbei aber, daß die **DOS/BIOS-Funktionen mit 16 Bit-Operanden** arbeiten und außerdem muß sichergestellt werden, daß der Assembler für den **Real-Mode 16Bit-Adressen** (und keine 32Bit-Adressen) erzeugt und dies geschieht am einfachsten durch die Reihenfolge der beiden folgenden Assembler-Direktiven

zuerst **.MODEL SMALL** (oder .MODEL TINY)
und dahinter **.486** (entspricht einer use16-Direktive = verwende 16Bit-Offset Adressen)

Hinweis: In allen Programmteilen einfach die 16Bit Register-Bezeichner durch vorangestelltes E auf 32 Bit Erweitern (AX → EAX) und im Eingabe-Unterprogramm eindez32 an einer Stelle korrigierend eingreifen !!!

2. System- und Hardware-Programmierung

1. Aufgabe Systemprogrammierung

Das vorgegebene **Unterprogramm-Paket eados.asm** (Ein- Ausgabe mit DOS-Funktionen des INT 21h) ist in eine **UP-Bibliothek eabios.asm** abzuwandeln, in der keinerlei DOS-Funktionen INT 21h, sondern nur noch BIOS-Funktionen verwendet werden.

Die Maus-Funktionen des INT 33h und die Grafikbildschirm-Initialisierung Fktn 0 des INT 10h in grafinit können dabei natürlich unverändert bleiben, aber die

Fktn 01 in getche,
 02 in putch,
 08 in getch und
 0Bh in kbhit des INT 21h müssen durch BIOS-Fktn ersetzt werden.

Da die **Bezeichner der Unterprogramme nicht verändert** werden sollen, braucht die INCLUDE-Datei eados.inc zum Text-Einfügen der EXTRN-Zuweisungen in das Hauptprogramm hierbei nicht verändert zu werden.

Die neu entstandene UP-Bibliothek ist mit dem Euklid-Programm des Praktikum 1.1 3. Aufgabe zu testen. Dazu muß die neue eabios-Objekt-Datei zum Hauptprogramm hinzugelinkt werden. (TLINK *****+eabios)

Skript: System-Funktionen und Systemunterlagen

2. Aufgabe Direkter Tastatur-Zugriff ohne Systemfunktionen

Das Unterprogramm-Paket eabios.asm ist in eine UP-Bibliothek **eadirekt.asm** umzuwandeln, die in einem ersten Schritt keinerlei Funktionen zur Konsol-Eingabe des DOS-INT 21h bzw Tastatur-Eingabe des BIOS-INT 16h verwendet, sondern **"direkt" auf die Tastatur zugreifen** soll

Die beiden INT16h-BIOS-Funktionen 0 (in getch) und 1 (in kbhit) in eabios.asm sollen hierfür durch ein Unterprogramm TASTATUR nachgebildet (emuliert) werden.

Dem UP TASTATUR werden dazu die gleichen Parameter wie dem INT 16h übergeben und die Ergebnisse sollen genauso zurückgeliefert werden.

Da in diesem Unterprogramm aber weder DOS noch BIOS-Funktionen verwendet werden dürfen, bietet sich hier für den Zugriff der **Tastatur-Ringpuffer** an, wobei aber nur das Auslesen des Ringpuffers mit eventuellem Nachführen des Lesezeigers zu programmieren ist.

Skript: Tastatur - Maus

Anhang

Euklidischer Algorithmus

berechnet den größten gemeinsamen Teiler zweier natürlicher Zahlen.

Die zwei Zahlen seien p und q .

1.Schritt: Dividiere p durch q ganzzahlig mit Rest r $r = p \bmod q$

2.Schritt: Wenn $r = 0$, dann ist q das Ergebnis

wenn $r \neq 0$, dann benenne q in p und r in q um
und wiederhole Schritt 1 und 2 solange, bis $r = 0$ wird

