

	FB Automatisierung und Informatik Laborpraktikum Mikrocomputertechnik & Assemblerprogrammierung		
	Versuch: MC&APIV	Thema: <p style="text-align: center;">Systemprogrammierung II</p>	
Protokollführer: Mitarbeiter 1: Mitarbeiter 2:	Seminargruppe: Versuchsgruppe: Laborplatz: Datum:		
Kolloquium:	Versuchstestat:	Protokoll:	Abschluss-Testat:

1. Versuchsziele

- Vertiefen der Kenntnisse über effektive Formen der Systemprogrammierung unter DOS
- Programmieren einfacher Peripheriefunktionen
- Erweiterung der Kenntnisse über das DOS-Dateisystem

2. Versuchsgrundlagen

- Befehlsliste I 80x86
- Assemblerdirektiven des TASM86
- DOS-Funktionen
- DOS-Datei-Konzept
- DOS-Handle-Konzept

3. Versuchsdurchführung

- 3.1 Legen Sie auf einer formatierten Diskette (LW A) die Datei „TEST.XYZ“ mittels Assembler an! Füllen Sie anschließend diese Datei mit dem Zeichensatz (00H-0FFH, aufsteigend)! Benutzen Sie dazu ein Speicher, in dem Sie die 256 Zeichen schreiben. Danach wird der Speicher in dtei geschrieben.
- 3.2 Legen Sie auf derselben Diskette die Datei „PROGRAMM.TXT“ an und speichern Sie alle Zeichen Ihres aktuellen Quelltextes in dieser Datei ab. Der Puffer darf eine maximale Größe von 256 Byte haben. Die Quelldatei muss mehr als 256 Zeichen haben.
- 3.3 Schreiben Sie ein Programm, dem eine beliebige Datei eines beliebigen Laufwerkes mittels Parameter übergeben wird. Geben Sie, sofern vorhanden, die ersten 256 Zeichen auf dem Bildschirm aus (DOS-Funktion). Im Fehlerfalle ist der Text „Datei -NAME- nicht gefunden!“ auszugeben. Der Parameter für das Programm wird im Debugger gesetzt (Menü Run, Eintrag „Arguments“).
- 3.4 Lesen Sie von dieser Diskette die ersten 64 Sektoren. Werten Sie die dort gespeicherten Informationen aus und erläutern Sie Boot-Sektor, FAT und Directory! Reservieren Sie genügend Speicher für die 64 Sektoren.
- 3.5 Löschen Sie von der Diskette die Datei „PROGRAMM.TXT“ und wiederholen Sie Aufgabe 3.4. Welche Informationen sind durch das Löschen verloren gegangen? Protokollieren Sie die Ergebnisse.
- 3.6 Schreiben Sie ein Programm zur Identifikation des CPU-Typs (32-Bit Modus) und geben Sie den ermittelten Typ auf dem Bildschirm aus (Hinweise: mwilhelm\download\asm).
Erstellen Sie dafür unbedingt eine neue Assembler-Datei (Quellcode *.asm).

Zusatzaufgabe:

Schreiben Sie ein Programm zur Binär/BCD-Konvertierung einer 16 Bit-Binärzahl (vorzeichenlos), die im Reg. CX steht. Das Ergebnis der Konvertierung soll im Reg. BL (0/ZT) und AX (T/H/Z/E) stehen.