

# Übungsblatt 2

Abgabe bis spätestens 9.11.2023 in Stud.IP

## Allgemeiner Hinweis

Für dieses Übungsblatt benötigt Ihr das in StudIP zur Verfügung gestellte Archiv ueb2-vorgabe.zip mit Vorgabedateien.

## Aufgabe 1 (3 Punkte)

Eure Aufgabe ist es, den Prozessbaum für das folgende Szenario grafisch darzustellen wie in der Vorlesung (Folienstapel ti2-03) gezeigt. Gebt genau eine Grafik ab, die den Zustand zu dem Zeitpunkt zeigt, der während der Ausführung von thunderbird am Ende dieses Szenarios herrscht. Die Wurzel des Teilbaums soll der von der Nutzerin gestartete Shellprozess sein (d. h. init, Login-Shell und der Windowmanager sollen nicht abgebildet werden). Beachtet auch, dass in dem Prozessbaum ausschließlich laufende oder angehaltene Prozesse enthalten sind. **Markiert außerdem die Prozesse, die zum Zeitpunkt dieser Momentaufnahme gerade im Vordergrund ausgeführt werden.**

Szenario:

Eine Nutzerin öffnet auf ihrem Desktop ein neues Fenster mit einer Unix-Shell und führt folgende Kommandos aus:

```
-> cd ti2
-> vim &
-> jobs
-> ping 134.102.218.18
```

Nun gibt sie Ctrl-z ein, gefolgt von:

```
-> emacs ueb2.tex
```

Danach drückt sie erneut Ctrl-z, und gibt dann folgendes ein:

```
-> bg
-> kill %1
-> sleep 3 &
-> bash
-> xterm &
-> firefox
```

Schließlich gibt sie folgendes im neu geöffneten X-Terminal ein:

```
-> thunderbird
```

## Aufgabe 2 (6 Punkte)

Im Archiv ueb2-vorgabe.zip findet Ihr unter aufgabe2 zwei Objektdateien, die auf dem Rechner x03 im Fachbereichsnetz aus C++-Quellcode erzeugt worden sind.

- Findet mit Hilfe von `nm` heraus, in welchem Modul die Funktion `main()` definiert ist. Dokumentiert dazu Eure Vorgehensweise.
- Beim Aufruf des Compilers mittels `gcc -v mage.o stat.o` erhaltet Ihr eine Reihe von Fehlermeldungen (undefined reference). Wodurch entstehen sie, das heißt, welcher Code kommt im Programm vor, für den dem Linker Informationen fehlen?
- Ergänzt die notwendige(n) Bibliothek(en) am Ende des `gcc`-Aufrufs (Kandidaten für Bibliotheken findet Ihr auf x03 in `/usr/lib/gcc/x86_64-linux-gnu/12/`). Um welche Bibliothek(en) handelt es sich? Wie würde man sie im Programmcode einbinden?
- Entfernt `mage.o` aus Eurem Aufruf. Welche Fehlermeldung bekommt Ihr? Wie sieht der dazu relevante Eintrag in der Symboltabelle der Objektdatei an dieser Stelle aus? Welches Problem ergibt sich dadurch für den Linker?
- Was verrät Euch die in d) genannte Fehlermeldung über den Programmcode in `mage.cc` und `stat.cc`?

**Bonus (1 Punkt):** Rekonstruiert, wie der Programmcode der Dateien `mage.cc`, `stat.cc` und eventuell außerdem benötigter Dateien aussehen müsste, damit die entsprechenden Symboltabellen sowie die Ausgabe des kompilierten Programms entstehen.

## Abgabe

Bis 23:59 Uhr am 9.11.2023 digital in StudIP. **Es gelten die vereinbarten Scheinbedingungen (siehe StudIP).** Bitte beachtet unsere ergänzenden Hinweise ebenda.

Ladet die abgaberelevanten Dateien in DoIT hoch.

Eure Ansätze und der gewählte Lösungsweg müssen nachvollziehbar sein. Achtet insofern auf eine saubere Dokumentation im Quelltext und im abgegebenen PDF-Dokument. Benennt alle von Euch verwendeten Quellen, auch Zusammenarbeit mit anderen Gruppen und verwendete Unterlagen aus früheren Jahrgängen.

Programmieraufgaben sind im Zweifel in C++20 zu entwickeln. Die Korrektheit der Lösung bzw. deren Grenzen sind grundsätzlich in der Abgabe nachzuweisen. Dies geschieht neben der Dokumentation des Programmcodes in den Quelldateien und zusätzlich in dem abgegebenen PDF-Dokument mit der Lösungsbeschreibung durch geeignete Tests, deren Auswahl und Eignung begründet werden müssen. Als **Referenzplattform** gelten die Linux-Rechner im Rechnerpool in MZH E0.