

2.2.F Funktionen/Funktionen benutzen – Ergänzungen und Bilder

2.2.F.1 Installation von `liner` als vorcompiliertes Binärprogramm

- a) Browser in Linux (!) öffnen
- b) Website <http://www.gyrator.de/material> anwählen
- c) Abschnitt „C2.2 Funktionen benutzen“ finden
- d) Dort Link „Software“ öffnen
- e) Dort Link „liner-1.09c (Verzeichnis)“ öffnen (kann auch 1.09d, 1.09e, 1.09f usw. sein)
- f) Dort Datei-Eintrag `liner` finden
- g) Die Datei herunterladen mit Klick auf rechter Maustaste (Kontextmenü) und Auswahl „Ziel Speichern unter ...“
- h) Speichern im persönlichen Verzeichnis
- i) Dasselbe (finden, herunterladen, speichern) wiederholen für die Dateien `liner.h` und `example1.c`
- j) Konsole öffnen
- k) Mit `ls -ltr` herausfinden, ob `liner`, `liner.h` und `example1.c` im persönlichen Verzeichnis vorhanden sind
- l) Die Datei `liner` ausführbar machen mit dem Befehl:

```
chmod 755 liner
```
- m) Das Programm `liner` ausführen:

```
./liner
```
- n) Es öffnet sich ein leeres weißes Fenster.
- o) Auf der Textkonsole kann man jetzt einen Graphikbefehl eingeben, z. B. :

```
R 50 50 100 100 1
```


Es müsste im weißen Fenster ein schwarzes Rechteck erscheinen
- p) Beenden des weißen Fensters mit `Q` oder `Strg` - `C`
- q) Administrator-Berechtigungen holen:

```
su
```
- r) Das Administrator-Passwort blind eingeben (es werden keine Punkte oder Sterne angezeigt)
- s) Die Eingabeaufforderung endet nun mit Raute # statt Dollarzeichen \$; das zeigt, dass man nun unbeschränkte Rechte auf dem System hat.
- t) Das Programm an die richtige Stelle kopieren:

```
cp liner /usr/local/bin
```
- u) Die Header-Datei an die richtige Stelle kopieren:

```
cp liner.h /usr/local/include
```
- v) Administrator-Berechtigungen wieder abgeben:

```
exit
```

 (oder `Strg` - `D`)
- w) Beispiel-Programm compilieren:

```
gcc example1.c
```

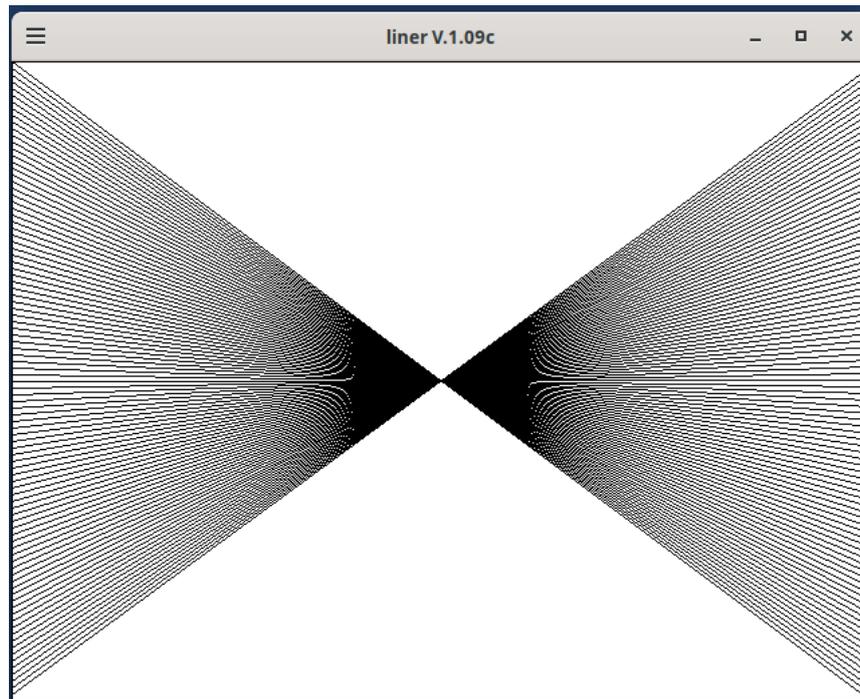


Abbildung 1: Ausgabe des Programms `example1`

- x) Beispiel-Programm ausführen:
`a.out`
Es erscheint ein Fenster wie in Abbildung 1.
- y) Anmerkung: Das Paket `libx11-6` muss installiert sein. Falls `liner` nicht läuft, kann es daran liegen, dass das Paket fehlt. Das kann man nachprüfen mit:
`dpkg -l libx11-6`
Falls die letzte Zeile mit `ii` beginnt, ist das Paket installiert:
`ii libx11-6:amd64 2:1.2.3-1+deb11u2 amd64 X11 client-side library`
Falls sie mit `un` beginnt, fehlt das Paket:
`un libx11-6 <keine> <keine> (keine Beschreibung vorhanden)`
In diesem Fall kann man es installieren:
`su` (Passwort blind eingeben)
`apt install libx11-6`
`exit`
- z) Anmerkung: Auf anderen Hardware-Plattformen muss man `liner` selbst compilieren. Dazu lädt man sich alle Datei aus dem Verzeichnis „`liner-1.09c` (Verzeichnis)“ herunter (am besten in ein neues Verzeichnis).
Dann ist es nötig, die Pakete `g++`, `make`, `libx11-6` und `libx11-dev` installiert zu haben. Das funktioniert wie oben beschrieben.
Zum Compilieren wechselt man in das Verzeichnis, in dem die heruntergeladenen Dateien liegen. Dann gibt man auf der Konsole ein:
`make`
`su` (Passwort wieder blind eingeben)
`make install`
`exit`