

```

In [11]: import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVC

# Definierte Datenpunkte und Labels
X = np.array([[1, 1], [-1, 1], [0, 0], [2, 1], [1, -2], [-2, 1], [3, 3]])
y = np.array([1, 1, 1, -1, -1, -1, -1]) # Labels: +1 für Klasse A, B, C und
-1 für Klasse D, E, F, G

# Erstelle das SVM-Modell mit einem linearen Kernel
svm_model = SVC(kernel='linear')
svm_model.fit(X, y)

# Extrahiere die Koeffizienten der Trennlinie (w und b)
w = svm_model.coef_[0]
b = svm_model.intercept_[0]

# Gib die Trennlinie in der Form "w1 * x1 + w2 * x2 + b = 0" aus
print(f"Trennlinie: {w[0]:.2f} * x1 + {w[1]:.2f} * x2 + {b:.2f} = 0")

# Bereich für x1 (damit die Linie in einem sinnvollen Bereich dargestellt wird)
x1_range = np.linspace(-3, 4, 100)

# Berechne x2-Werte für die Trennlinie
x2_range = -(w[0] / w[1]) * x1_range - (b / w[1])

# Plot der Datenpunkte und der Trennlinie
plt.figure(figsize=(8, 6))

# Plot der Datenpunkte
plt.scatter(X[y == 1, 0], X[y == 1, 1], color='red', label='Klasse +1')
plt.scatter(X[y == -1, 0], X[y == -1, 1], color='blue', label='Klasse -1')

# Plot der Trennlinie
plt.plot(x1_range, x2_range, 'k--', label='Trennlinie')

# Plot der Support-Vektoren
plt.scatter(svm_model.support_vectors[:, 0], svm_model.support_vectors[:, 1],
            s=100, facecolors='none', edgecolors='k', label='Support-Vektoren')

# Achsenbeschriftungen und Titel
plt.xlim(-3, 4)
plt.ylim(-3, 4)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
plt.title('SVM-Trennlinie mit linearem Kernel')
plt.legend()
plt.grid(True)
plt.show()

```

Trennlinie: $-0.63 * x_1 + 0.25 * x_2 + 0.13 = 0$

