Beispiel 1:

1. Erweitern Sie das „Streaming-Backup“ aus der 1. Übung:
   Schreiben Sie zusätzlich am Beginn des Archivs eine Kennung, die das Archive-File als Archiv ihres „Backup-Typs“ kennzeichnet. Diese Kennung besteht aus Ihrem Familiennamen (Zeichenfolge OHNE Null-Terminierung!!) und Ihrem Geburtsdatum (als Unix Zeitwert). Damit beginnt die Datei z.B. so:

‚H‘	‚u‘	‚b‘	‚b‘	‚e‘	‚r‘	T1	T2	T3	T4	A	A	A	….	….	….

   T1 bis T4 sind die 4 Bytes, die Ihr Geburtsdatum repräsentieren (Verwende Sie zB mktime() um den Wert zu erzeugen). Diese Kennung ist damit bei jedem Studenten anders und potentiell unterschiedlich lange. AAA…. ist der Beginn des eigentlichen Archivs, also zB der Name oder Inode des 1. Eintrags.

2. Adaptieren Sie das dazugehörige Restore-Programm folgendermaßen:

   a. Zuerst ist die Archivkennung zu Prüfen und Familienname und Geburtsdatum, die dort drin stehen als Info auszugeben. Sollte es sich nicht um Ihre Daten handeln ist mit Fehlermeldung abzubrechen.
   b. Versuchen Sie die Dateien mit ihren original Zugriffsrechten samt Owner und Group Information zu erzeugen. Die nötigen Infos nehmen Sie aus den gespeicherten Inode-Infos am Archiv. Sollte das nicht gelingen (zB fehlende Rechte), geben Sie ein Warning aus.
   c. Wenn im gespeicherten Inode der Link-Counter bei Regular Files größer als 1 ist, geben Sie diesen aus samt einem Warning, dass alle Hardlinks zu einem neuen File führen.
   d. Sie müssen nur Dateien und Verzeicnisse wiederherstellen können. Für alle anderen Dateitypen geben Sie aus, dass dieser File-Typ derzeit nicht unterstützt wird.

Beispiel 2:

Programmieren Sie Ihre eigene Shell, die folgendes leistet:

a) Die Shell muss die folgende (in der Übung zum Teil schon gemeinsam erledigten) interne Kommandos beherrschen:
   a. cd
   b. pwd
   c. id *[gibt uid, euid, gid, egid (samt Namen, falls verfügbar) aus]*
   d. exit
   e. umask
   f. printenv *[gibt die Environment-Variablen des Shell-Prozesses aus]*

b) Die Shell muss die Environment Variable PATH ändern können. Sehen sie dafür einen eigenen Befehl *setpath* vor, mit dem das möglich ist. Beim Aufruf externer Befehle ist dieser natürlich zu verwenden.

c) Ein Shell-interner Befehl ‚info' gibt folgende Infos über die Shell aus:
   Shell von: *Ihr Familienname und Ihre Matrikelnummer*
   PID: *PID des Shell Prozesses*
   Läuft seit: *Datum und Uhrzeit, wann die Shell gestartet wurde*

d) Der Prompt der Shell enthält das Workingdirectory und den Rechnernamen (verwenden Sie den Nodename aus uname()). Wenn sich das Workingdirectory ändert, muss sich natürlich auch der Prompt ändern.

e) Die Shell muss Vordergrund- und Hintergrundjobs starten können. Dabei ist darauf zu achten, dass die Signale SIG_INT und SIG_QUIT im Vordergrund zum Terminieren des Vordergrundjobs aber nicht zum Terminieren eines laufenden Hintergrundjobs oder gar der Shell selbst führen.

Beispiel 3:

Machen Sie einen detaillierten Entwurf für die Erweiterung Ihrer Shell um folgendes Feature:
Bearbeiten Sie je nach letzter Stelle Ihrer Matrikelnummer die Variante:

0 oder 5: a.
1 oder 6: b.
2 oder 7: c.
3 oder 8: d.
4 oder 9: e.

e. Wildcards
f. Unterstützung von Kommando-History
g. Unterstützung von Shell-Variablen
h. Implementieren des trap-Kommandos (Signalhandler in der Shell)
i. Ein- und Ausgabeumleitung

Überlegen Sie dazu:
- Welche zusätzlichen Befehle müssen Sie vorsehen?
- Ändert sich gravierendes an der Grundstruktur Ihrer Shell?
- Welche Informationen müssen dafür in der Shell abgefragt oder aufbewahrt werden?
- Welche zusätzlichen Systemcalls müssen/wollen Sie verwenden?
- Wo in der Shell muss was geändert / eingebaut werden?

Der Programmentwurf bzw. Änderungsentwurf ist in gutem Deutsch oder Englisch zu verfassen und als PDF mit den restlichen Übungsaufgaben abzugeben. Bei der Beschreibung von einzubauenden Codestücken können Sie Pseudocode verwenden.