Inhalt

[1. Anforderungen an das Script 2](#_Toc155892408)

[2. Technische Beschreibung 3](#_Toc155892409)

[3. Testen 4](#_Toc155892410)

[3.1. Testfall „Full Backup“ 4](#_Toc155892411)

[4. Testergebnisse 6](#_Toc155892412)

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Datum | Autor | Bemerkungen |
| 0.1 | 08.01.204 | FEC | Initiale Version |

# Anforderungen an das Script

* Was sind die Aufgaben des Scripts?

Es sollte ein automatisiertes Backup erstellt werden, das regelmäßig ausgeführt wird. Hierbei ist vorgesehen, einmal pro Woche ein "Full Backup" durchzuführen, andernfalls soll täglich ein "Incremental Backup" erfolgen. Sollte am Zielort noch kein Backup vorhanden sein, ist zunächst ein "Full Backup" durchzuführen. Das Zielverzeichnis sollte auf einer externen Festplatte oder einem USB-Stick liegen. Im Falle der Nichtexistenz des Zielverzeichnisses soll eine entsprechende Fehlermeldung ausgegeben werden.

* Welche Funktionen sollen zur Verfügung gestellt werden?

Das Speichermedium sollte selbst bestimmt werden. Dabei überprüft das Programm selbst, ob das Speichermedium schon verwendet wurde oder nicht. Bei einem «Full Backup» kann selbst entschieden werden, wie viele «Full Backups» jeweils auf dem Speichermedium gesichert werden sollte. Zudem kann auch selbst bestimmt werden, wo die Output Datei gespeichert wird. Die Output Datei ist die Datei, wo Rückmeldungen (Fehlermeldungen oder Erfolge) festgehalten werden. Eine identische Version wird jedoch auch auf dem USB-Stick direkt gespeichert.

* Welche Ausnahmesituationen können auftreten, und wie wird mit denen umgegangen?

Eine Ausnahmesituation könnte sein, falls das Speichermedium nicht mehr genügend Speicherplatz mehr hat. Dies sollte jedoch nicht der Fall sein, da selbst bestimmt werden kann, wie viele «Full Backups» gespeichert werden sollte. Wäre dies trotzdem der Fall, wird eine Fehlermeldung ausgegeben, dass das Speichermedium nicht mehr genügend Speicherplatz mehr hat.

Eine weitere Ausnahmesituation könnte sein, dass das Speichermedium nicht vorhanden ist. Dabei wird auch eine Fehlermeldung ausgegeben.

# Technische Beschreibung

* Wie ist der Ablauf des Scripts? Programmablaufplan

[PAP-TIP2.pdf](https://hsluzern-my.sharepoint.com/:b:/r/personal/colin_felber_stud_hslu_ch/Documents/PAP-TIP2.pdf?csf=1&web=1&e=iZQXC6)

* Was sind die Unterschiede zwischen dem vollen und dem inkrementellen Backup?

Bei einem «Full Backup» werden alle Daten von den Benutzern gespeichert (/home/\*). Dabei werden diese durch eine ZIP Datei archiviert und komprimiert. Bei jedem «Full Backup» werden alle Daten von Grund auf neu gesichert. Sie bieten eine vollständige Wiederherstellung der Daten, sind jedoch zeitaufwändig und erfordern mehr Speicherplatz. Dabei kann selbst definiert werden, wie viele Backups gespeichert werden sollen. Bei «Incremental Backup» sichert nur die Daten, die seit dem letzten Backup hinzugefügt oder geändert wurden. Zudem wird überprüft, welche Daten gelöscht wurden, welche darauf hin auch gelöscht werden. Bei einem «Incremental Backup» wird nur der Unterschied zu einem vorherigen Backup gespeichert. «Incremental Backup» sind in der Regel schneller.

* Wie wird unterschieden welche Art von Backup ausgeführt wird?¨

Die wird mithilfe von einer Positionsparameter definiert. Dabei kann angeben werden, ob ein 0 = «Full Backup» durchgeführt werden sollte oder mit der 1 = «Incremental Backup».

* Wie wird das Script in das System integriert? Wo ist das Script abgespeichert?

Das Script wurde im System integriet, indem bei /usr/local ein Ordner erstellt wurde. Dabei hat nur die Gruppe «admin» zugriff auf diesen Ordner. Dabei wurde eine Benutzergruppe «admin» erstellt mit den Benutzern admin & backu. Somit hat jeder admin User zugriff auf diesen Ordner und ist nicht bei einem einzelnen User abgelegt.

* Was passiert wenn kein USB-Stick angeschlossen ist?

Auf einer «.log» Datei wird eine Fehlermeldung erscheinen, dass kein Speichermedium vorhanden ist. Falls dies der Fall ist, wird das Skript per «Exit 1» verlassen.

# Testen

* Das Script muss an Anlehnung an Kapitel 15 aus dem Modul 122 auch getestet werden.

Beim Kapitel 15 wird nichts übers Testen festgehalten. Jedoch habe ich einen Whitebox-Test durchgeführt. Das Bedeute, ich habe den Code für mich selbst nochmals studiert und Kleinigkeiten angepasst. Dies Testergebnisse findet man unten:

* Was für Vorbedingungen müssen für die Testfälle erfüllt sein?

Um ein Skript erfolgreich zu testen, müssen mehrere Punkte erfüllte sein:

* Korrekte Umgebungseinrichtung, sodass sichergestellt werden kann, dass nicht nur das Programm auf meiner Umgebung funktioniert, sondern auch in der «live» Umgebung.
* Berechtigungen, sodass das Skript nicht Fehler herausgibt, weil das Skript keine Berechtigungen auf die Daten hat.
* Mindestens 2 Testfälle werden erwartet. Zum Beispiel: Feststellen, dass ein volles Backup, notwendig ist. Was passiert, wenn kein Memory Stick vorhanden ist?

## Testfall „Full Backup“

|  |  |
| --- | --- |
| **Name** | Full Backup |
| **Vorbedingung** | * Es ist Mittwoch * der cron daemon wird ausgeführt um 18:13 (Zeit selbst definierbar) * Ein USB-Stick mit dem Namen „usbbackup“ ist in den Ordner /media/admin1/ gemountet. |
| **Ablauf** | 1. Das Skript überprüft, ob das Speichermedium vorhanden ist. Ist in diesem Fall vorhanden. 2. Das Skript überprüft, ob die Dateistruktur auf dem USB-Stick vorhanden ist. Ist in diesem Fall vorhanden. 3. Es wird überprüft, wie viele «Full Backups» vorhanden sind und wie viel gewünscht sind. In diesem Fall sind bereits 5 Backups vorhanden und es sind nur 5 Backups erwünscht. 4. Das Skript löscht die älteste Datei und erstellt ein neues Backup direkt auf dem USB-Stick. 5. Der Erfolg / Misserfolg vom Backup wird auf beiden Output Dateien festgehalten. |
| **Erwartetes Ergebnis** | Das Verzeichnis „/home/\*“ wurde auf den USB-Stick direkt als ZIP kopiert. Dabei sollte die ZIP-Datei die aktuelle Uhrzeit haben. |
| **Bilder:** | Ausgabe auf dem USB-Stick.    Ausgabe, bei dem der Pfad definiert wurde: |

|  |  |
| --- | --- |
| **Name** | Incremental Backup |
| **Vorbedingung** | * Es ist Mittwoch * der cron daemon wird ausgeführt um 18:26 (Zeit selbst definierbar) * Ein USB-Stick mit dem Namen „usbbackup“ ist in den Ordner /media/admin1/ gemountet. |
| **Ablauf** | 1. Das Skript überprüft, ob das Speichermedium vorhanden ist. Ist in diesem Fall vorhanden. 2. Das Skript überprüft, ob die Dateistruktur auf dem USB-Stick vorhanden ist. Ist in diesem Fall vorhanden. 3. Das «inkremental Backup» wird durchgeführt. 4. Der Erfolg / Misserfolg vom Backup wird auf beiden Output Dateien festgehalten. |
| **Erwartetes Ergebnis** | Das Verzeichnis „/home/\*“ wurde auf den USB-Stick kopiert und als ZIP gepackt. Dabei sollte die ZIP-Datei die aktuelle Uhrzeit haben. |
| **Bilder:** | Ausgabe auf dem USB-Stick.    Ausgabe, bei dem der Pfad definiert wurde: |

|  |  |
| --- | --- |
| **Name** | Kein Speichermedium vorhanden |
| **Vorbedingung** | * Es ist Mittwoch * der cron daemon wird ausgeführt um 18:37 (Zeit selbst definierbar) * Kein Speichermedium ist vorhanden. |
| **Ablauf** | 1. Das Skript überprüft, ob das Speichermedium vorhanden ist. Ist in diesem Fall nicht vorhanden. 2. Der Misserfolg vom Backup wird auf beiden Output Dateien festgehalten. |
| **Erwartetes Ergebnis** | Es sollten Fehlermeldungen ausgegeben werden. |
| **Bilder:** | Ausgabe, bei dem der Pfad definiert wurde: |

# Testergebnisse

* Zusammenfassung der Testergebnisse:  
  Wann hat wer gestestet?

Das Testen hat nur Colin Felber übernommen. Dabei wurde schon vieles während dem Programmieren getestet. Dies war nur noch ein Test am Schluss, ob alles wirklich so funktioniert wie es sollte. Dabei wurde alles per Cron-Demon ausgeführt. Um sicherzustellen, dass es funktioniert. Aus Zeitlichen Gründen, war es nicht möglich eine Woche abzuwarten, um ein Wöchentliches «Full Backup» zu testen.

Stimmte das tatsächliche Ergebnis mit dem erwarteten Ergebnis überein?

Beim Testen wurde noch ein Fehler gefunden. Zwar beim Erkennen von keinem USB-Stick hat das Skript unzähliche Fehlermeldungen ausgegeben. Aus Zeitlichen Gründen, wurde eine einfache Lösung gewählt. Zwar kann nicht mehr selbst im Skript direkt ein neues Verzeichnis ausgewählt werden. Sondern das Skript muss wieder neu gestartet werden. Dies sollte jedoch nicht so ein Problem sein, da der Befehl per History schnell wieder geholt werden kann.

Vorher:

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Nachher:

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung