Abschlusstest
Linux Advanced

Bearbeitungszeit:	60 Minuten
Zugelassenes Hilfsmittel:	selbst erstellte Befehlsreferenz
Datum:	28.03.25

Name: _____

Klasse: _____

Wird vom Korrektor ausgefüllt:

Punkte: _____ von 100

Unterschrift Korrektor: _____

Frage 1 [5 Punkte]

Welcher Befehl zeigt die IP-Adresse(n) deines Systems an?

Kreuzen Sie die richtige Antwort an (nur eine ist richtig).

ifconfig -d	
ip address show	X
dig	
ping	
show ip	

Frage 2 [5 Punkte]

Welcher Nice-Wert gibt einem Prozess die höchste Priorität?

Kreuzen Sie die richtige Antwort an (nur eine ist richtig).

9	
-10	
20	
-20	X
19	

Frage 3 [5 Punkte]

Was ist /dev/null

Kreuzen Sie die richtige Antwort an (nur eine ist richtig).

Ein spezieller Prozess	
Ein versteckter Benutzer	
Eine „Datenmülltonne“, in der Ausgaben verschwinden	X
Der Kernel-Zufallszahlengenerator	
Eine normale Datei	

Frage 4 [5 Punkte]

Welche Aussagen zu systemd treffen zu?

Kreuzen Sie die richtigen Antworten an (mindestens drei sind richtig; aber nicht alle).

systemd ersetzt klassische Init-Systeme wie SysVinit	X
Eine .service-Datei beschreibt einen Dienst	X
systemd startet ausschließlich grafische Benutzeroberflächen	
systemd kann anhand von Units Netzwerk, Timer, Mounts und Services konfigurieren	X
systemd kann nicht zum loggen genutzt werden	

Frage 5 [5 Punkte]

Was lässt sich mit journalctl anzeigen?

Kreuzen Sie die richtigen Antworten an (mindestens zwei sind richtig; aber nicht alle).

Festplattennutzung	
Logs eines bestimmten Dienstes	X
Die CPU Auslastung	
Aktive Prozesse	
Meldungen des letzten Bootvorgangs	X

Frage 6 [5 Punkte]

Welche dieser Dateien enthalten System-Logs?

Kreuzen Sie die richtigen Antworten an (mindestens zwei sind richtig; aber nicht alle).

/var/log/auth.log	X
/etc/resolv.conf	
/var/log/wtmp	X
/usr/bin/logger	
/usr/bin/log	

Frage 7 [5 Punkte]

Welche Aussagen zum Shellskripting sind korrekt?

Kreuzen Sie die richtigen Antworten an (mindestens drei sind richtig; aber nicht alle).

Der Befehl echo gibt immer auf stderr aus	
Variablen werden mit = zugewiesen, z. B. NAME=Alex	x
#!/bin/bash gibt an, welche Shell das Skript interpretieren soll	x
Shellskripte benötigen IMMER eine Dateiendung .sh	
Mit chmod +x wird ein Skript ausführbar	x

Frage 8 [5 Punkte]

Welche Aussagen zu SSH sind korrekt?

Kreuzen Sie die richtigen Antworten an (mindestens drei sind richtig; aber nicht alle).

Der Dienst heißt sshd	x
SSH ist standardmäßig unverschlüsselt	
Mit scp kann man Dateien übertragen	x
Die Datei /etc/ssh/sshd_config konfiguriert den Server	x
SSH ist nur innerhalb eines LANs möglich	

Frage 9 [6 Punkte]

Was geben folgende Befehle aus – und was bedeuten die jeweiligen Variablen?

```
echo "${HOME}"
echo "${USER}"
echo "${SHELL}"
```

${HOME} Das Heimatverzeichnis des Aktuellen Benutzers
${USER} Der Benutzername des Aktuell angemeldeten Benutzers
${SHELL} Die Shell etwa /bin/sh oder /bin/bash des Aktuell angemeldeten Benutzers

Frage 10 [6 Punkte]

Wie kannst du verhindern, dass Logdateien mit der Zeit immer größer werden und irgendwann den gesamten Systemspeicher belegen?

Um zu verhindern, dass Log-Dateien auf einem dauerhaft laufenden Server (z. B. 24/7) mit der Zeit immer größer werden und letztlich den gesamten Speicherplatz belegen, kann das Tool logrotate verwendet werden. Dieses löscht alte Logs automatisiert in einem benutzerdefinierten Intervall – gesteuert durch einen systemd-Timer. Je nach Konfiguration bleiben dabei die Log-Dateien der letzten N Wochen oder Monate erhalten und einsehbar. Aus Datenschutzgründen ist eine solche Rotation oft notwendig, etwa um die DSGVO-Konformität bei IP-loggenden Webservern in der EU sicherzustellen.

Frage 11 [6 Punkte]

Ein Shellskript wird wie folgt aufgerufen: ./mein-skript.sh hallo welt

Was geben die folgenden Variablen im Skript aus?

echo "${1}" hallo

echo "${2}" welt

echo "${@}" hallo welt => alles array z.b. für eine for-schleife

echo “${*}” hallo welt => als zusammenhängender String

echo "${#}" Anzahl der Argumente im Array

Frage 12 [8 Punkte]

Du willst regelmäßig den Befehl echo "Hallo Welt" zur vollen Stunde ausführen. Nenne zwei Möglichkeiten, das zu erreichen, und beschreibe grob, wie du sie einrichtest und wie sie sich unterscheiden (Vor- und Nachteile).

Um den Befehl echo "Hallo Welt" regelmäßig zur vollen Stunde auszuführen, gibt es zwei gängige Möglichkeiten: die Verwendung von Cron oder eines systemd-Timers.

Die einfachere Methode ist ein Cronjob. Dazu öffnet man den Crontab-Editor mit crontab -e und fügt folgende Zeile ein: 0 * * * * echo "Hallo Welt". Damit wird der Befehl stündlich zur Minute 0 – also zur vollen Stunde – ausgeführt. Diese Lösung ist schnell eingerichtet und auf nahezu jedem Linux-System verfügbar. Sie eignet sich besonders gut für einfache, regelmäßig wiederkehrende Aufgaben. Allerdings bietet Cron von Haus aus kein Logging, keine Fehlerbehandlung und keine Integration in das moderne systemd-Ökosystem. Für komplexere Aufgaben ist es daher weniger flexibel.

Die zweite Möglichkeit ist ein systemd-Timer. Dafür erstellt man zunächst eine Service-Datei, z. B. /etc/systemd/system/hallo.service, die den gewünschten Befehl ausführt. In diesem Fall enthält sie den Eintrag ExecStart=/bin/echo "Hallo Welt". Anschließend legt man eine passende Timer-Datei wie /etc/systemd/system/hallo.timer an, in der man mit OnCalendar=hourly festlegt, dass der Dienst zur vollen Stunde gestartet wird. Durch Persistent=true wird sogar sichergestellt, dass verpasste Ausführungen nachgeholt werden (z. B. wenn der Rechner ausgeschaltet war). Nach dem Aktivieren mit sudo systemctl enable --now hallo.timer läuft der Timer automatisch im Hintergrund. Diese Methode bietet eine tiefere Integration ins System, automatische Protokollierung über journalctl und mehr Kontrolle über Ausführung, Fehlerbehandlung und Abhängigkeiten. Der Nachteil ist, dass die Einrichtung aufwendiger ist als bei einem einfachen Cronjob.

Frage 13 [8 Punkte]

Was bewirkt der folgende Befehl?

dmesg | grep error >> log

Erkläre dabei kurz, was eine | und >> macht und was bei dmesg | grep error > log **anders wäre**

Der Befehl dmesg | grep error >> log dient dazu, alle Kernelmeldungen, die das Wort „error“ enthalten, auszugeben und in eine Datei namens log anzuhängen. Dabei passiert Folgendes im Detail:

Zunächst liefert dmesg eine Liste aller aktuellen Kernelmeldungen, also Systemnachrichten, die beim Start und während des Betriebs des Systems entstehen. Der senkrechte Strich | ist die sogenannte Pipe. Sie leitet die Ausgabe des linken Befehls (in diesem Fall dmesg) direkt als Eingabe an den rechten Befehl (grep error) weiter. grep filtert dann alle Zeilen heraus, die das Wort „error“ enthalten. Anschließend sorgt der Operator >> dafür, dass diese gefilterten Zeilen am Ende der Datei log angehängt werden – falls die Datei nicht existiert, wird sie automatisch erstellt.

Im Unterschied dazu würde der Befehl dmesg | grep error > log nicht anhängen, sondern die Datei log bei jeder Ausführung überschreiben. Das bedeutet: Mit >> wird der bestehende Inhalt der Logdatei beibehalten und ergänzt, während > bei jeder Ausführung die Datei neu schreibt und vorherige Inhalte verwirft.

Frage 14 [8 Punkte]

Du arbeitest in einem Terminal und führst den Befehl sleep 300 aus.
Plötzlich fällt dir ein, dass du noch andere Dinge tun musst.
Beschreibe Schritt für Schritt:

a.) Wie du den laufenden Befehl im Hintergrund weiterlaufen lässt.

b.) Wie du dir alle laufenden Jobs anzeigen lässt.

c.) Wie du den Job wieder in den Vordergrund holst.

d.) Was passiert, wenn du das Terminal schließt, während der Job läuft.

e.) Wie du einen Befehl, von Anfang an im Hintergrund starten kannst.

a.) Laufenden Befehl in den Hintergrund, ohne ihn zu beenden

1. **Drücke `Strg + Z`** (stoppt den aktuell laufenden Befehl temporär)
2. **Gib `bg` ein und drücke Enter** (Befehl wird im Hintergrund fortgesetzt)

b.) Alle laufenden Jobs anzeigen

1. **Gib den Befehl `jobs` ein.**

c.) Wie du den Job wieder in den Vordergrund holst

1. **Gib den Befehl `fg` ein** (zuletzt gestartete Job in den Vordergrund)
2. Wenn du mehrere Jobs hast Verwende fg JOBID

d.) Terminal schließen, während ein Job im Hintergrund läuft

1. **Standardmäßig wird der Job beendet, sobald du das Terminal schließt.**
   → Hintergrundjobs sind an das Terminal gebunden.
2. **Ausnahme: Wenn der Job mit nohup gestartet wird.**

e.) Von Anfang an einen Befehl direkt im Hintergrund starten kannst

- **Beispiel:**

```
sleep 300 &
```

Frage 15 [9 Punkte]

Ein Prozess reagiert nicht mehr. Welche Möglichkeiten hast du, ihn zu identifizieren und sauber oder sofort zu beenden? Nenne und erkläre mindestens 3 Befehle.

Mit ps aux | grep <Name> lässt sich gezielt nach einem Prozess suchen und seine PID (Prozess-ID) herausfinden. Alternativ zeigt htop eine übersichtliche, interaktive Liste aller Prozesse, aus der man den betroffenen Prozess direkt auswählen und beenden kann. Zum sauberen Beenden nutzt man kill <PID>, das ein freundliches SIGTERM-Signal sendet. Reagiert der Prozess nicht, hilft kill -9 <PID>, das den Prozess sofort mit SIGKILL beendet. Diese Variante sollte nur im Notfall verwendet werden, da sie keine Aufräumarbeiten erlaubt.

Frage 16 [9 Punkte]

Beschreibe, was der nachfolgende Codeblock in Bezug auf die Menge und den Inhalt der bereitgestellten Kommandozeilenargumente tut.

Was passiert, wenn ich das Script mit ./mein-script help aufrufe?

```
while [[ ${#} -gt 0 ]]; do
    case "${1}" in
        --help)
            echo "HILFE"
            ;;
        --debug)
            echo "DEBUG"
            ;;
        *)
            echo "ERROR"
            ;;
    esac
    shift
done
```

Der Codeblock verarbeitet alle übergebenen Kommandozeilenargumente einzeln in einer Schleife. Dabei prüft er mit case, ob das aktuelle Argument --help, --debug oder etwas anderes ist. Für bekannte Argumente gibt es eine entsprechende Ausgabe („HILFE“ oder „DEBUG“), unbekannte führen zu „ERROR“. Mit shift wird das jeweils erste Argument entfernt, sodass die Schleife weiterarbeiten kann, bis keine Argumente mehr übrig sind.

Wenn das Skript mit ./mein-script help aufgerufen wird, erkennt der Code help nicht als gültige Option (es fehlt das --), daher wird „ERROR“ ausgegeben.