Bash ???????

??? 1: ???????? ????? ???????

Создайте файл с расширением `.sh`. Первая строка всегда должна быть "shebang" — `#!/bin/bash`. Он указывает системе, какой интерпретатор использовать для выполнения скрипта.

```
#!/bin/bash
#Это комментарий
echo "Привет, мир!"
```

??? 2: ????????????? ???? ?? ??????????

По умолчанию текстовый файл не является исполняемым. Используйте команду `chmod +x`, чтобы это исправить.

```
chmod +x my_script.sh
```

??? 3: ?????? ???????

Для запуска укажите путь к файлу. Если вы находитесь в том же каталоге, используйте `./`.

```
./my_script.sh
```

????????????? ? ??????? Cron

Cron — это планировщик задач. Для редактирования задач текущего пользователя используется команда `crontab -e`. Каждая строка имеет формат: `МИНУТА ЧАС ДЕНЬ_МЕСЯЦА МЕСЯЦ ДЕНЬ_НЕДЕЛИ /путь/к/команде`.

- `* * * * *` — Каждую минуту.
- `0 * * * *` — Каждый час, в 00 минут.
- `0 */4 * * *` — Каждые 4 часа.
- `5 3 * * *` — Каждый день, в 3:05 ночи.
- `0 4 * * 0` — Каждое воскресенье, в 4:00 утра (0 - воскресенье).

Пример записи для выполнения скрипта ежедневно в 2:30 ночи:

```
30 2 * * * /home/user/scripts/backup.sh >/dev/null 2>&1
```

Часть `>/dev/null 2>&1` перенаправляет весь вывод (стандартный и ошибки) в "никуда", чтобы cron не отправлял вам письма по каждому запуску.

?????????? ??????? ????????

1. ????????? ??????????? ????????

Создает .tar.gz архив указанного каталога с датой в имени файла. Идеально для ежедневного бэкапа сайтов или важных документов.

```
#!/bin/bash
#Каталог, который нужно архивировать
SOURCE_DIR="/var/www/my-site"
# Каталог для хранения бэкапов
BACKUP_DIR="/mnt/backups/sites"
# Имя файла бэкапа с датой
BACKUP_FILENAME="my-site_$(date +%Y-%m-%d).tar.gz"
#Создание архива
tar -czf "$BACKUP_DIR/$BACKUP_FILENAME" -C "$(dirname "$SOURCE_DIR")" "$(basename
"$SOURCE_DIR")"
echo "Бэкап $BACKUP_FILENAME успешно создан."
```

2. ?????????? ????????? ????????????

Проверяет использование диска в указанном разделе. Если использование превышает порог, отправляет письмо на почту.

```
#!/bin/bash
#Порог в процентах
THRESHOLD=90
#Раздел для проверки
PARTITION="/"
#Email для оповещений
ADMIN_EMAIL="admin@example.com"
#Получаем текущее использование в процентах (без знака %)
CURRENT_USAGE=$(df "$PARTITION" | grep -v Filesystem | awk '{print $5}' | sed 's/%//g')
if [ "$CURRENT_USAGE" -gt "$THRESHOLD" ]; then
  echo "ВНИМАНИЕ: На сервере $(hostname) заканчивается место на диске ($PARTITION). Занято:
$CURRENT_USAGE%." | mail -s "Disk Space Alert" "$ADMIN_EMAIL"
fi
```

3. ???????? ? ?????????? ???????

Проверяет, активен ли указанный сервис (например, nginx). Если нет, пытается его перезапустить и отправляет уведомление.

```
#!/bin/bash
SERVICE_NAME="nginx"
ADMIN_EMAIL="admin@example.com"
if ! systemctl is-active --quiet "$SERVICE_NAME"; then
    echo "Сервис $SERVICE_NAME не активен! Попытка перезапуска..."
    systemctl restart "$SERVICE_NAME"
    sleep 2
    if systemctl is-active --quiet "$SERVICE_NAME"; then
        echo "Сервис $SERVICE_NAME успешно перезапущен." | mail -s "Service Restart:
$SERVICE_NAME" "$ADMIN_EMAIL"
    else
        echo "НЕ УДАЛОСЬ перезапустить сервис $SERVICE_NAME!" | mail -s "CRITICAL: Service
Down: $SERVICE_NAME" "$ADMIN_EMAIL"
    fi
fi
```

4. ??????? ?????? ???-??????

Находит и удаляет файлы, которые не изменялись более N дней в указанном каталоге. Помогает освободить место.

```
#!/bin/bash
#Каталог с логами
LOG_DIR="/var/log/my_app"
#Удалять файлы старше N дней
DAYS=30
find "$LOG_DIR" -type f -name "*.log" -mtime +"$DAYS" -delete
echo "Старые логи в $LOG_DIR удалены."
```

5. ??????????? ? ????? ?????????? IP

Полезно для домашних серверов с динамическим IP. Скрипт проверяет текущий IP и, если он изменился с последнего запуска, отправляет новый IP на почту.

```
#!/bin/bash
#Файл для хранения последнего известного IP
IP_FILE="/home/user/.last_ip"
```

```
ADMIN_EMAIL="myemail@example.com"
#Получаем текущий IP
CURRENT_IP=$(curl -s ifconfig.me)
if [ ! -f "$IP_FILE" ]; then
    echo "$CURRENT_IP" > "$IP_FILE"
fi
LAST_IP=$(cat "$IP_FILE")
if [ "$CURRENT_IP" != "$LAST_IP" ]; then
echo "Новый IP-адрес: $CURRENT_IP" | mail -s "IP Address Changed" "$ADMIN_EMAIL"
echo "$CURRENT_IP" > "$IP_FILE"
fi
```

6. ????? ???????? ?????? ?? SSH

Отправляет на почту список последних успешных входов по SSH. Помогает отслеживать, кто и когда заходил на сервер.

```
#!/bin/bash
ADMIN_EMAIL="security@example.com"
(echo "Последние 10 успешных входов по SSH на $(hostname):"; echo; last -n 10) | mail -s "SSH Login Audit" "$ADMIN_EMAIL"
```

7. ???????? "??????" ????????? ???????

Собирает основную информацию о состоянии системы (нагрузка, память, диск) и сохраняет в лог-файл. Полезно для ретроспективного анализа проблем.

```
#!/bin/bash
SNAPSHOT_FILE="/var/log/system_snapshots.log"
{
    echo "====== SNAPSHOT $(date) ======";
    echo "--- Uptime & Load Average ---";
    uptime;
    echo;
    echo "--- Memory Usage (MB) ---";
    free -m;
    echo;
    echo "--- Disk Usage ---";
    df -h;
    echo "==================================\n";
```

```
} >> "$SNAPSHOT_FILE"
```

8. ???????? ????????????? ??????

Пример: заменяет пробелы на знаки подчеркивания во всех именах файлов в текущем каталоге. Исправлено для предотвращения перезаписи существующих файлов.

```
#!/bin/bash
#Перебираем все файлы и папки в текущем каталоге
for file in *; do
    # Проверяем, есть ли пробелы в имени
    if [[ "$file" == *" "* ]]; then
        # Заменяем пробелы на подчеркивания
        new_name=$(echo "$file" | tr ' ' '_')
        # ПРОВЕРКА: не перезаписать существующий файл
        if [ ! -e "$new_name" ]; then
            mv "$file" "$new_name"
            echo "Переименовано: '$file' -> '$new_name'\n"
        else
            echo "ОШИБКА: Файл '$new_name' уже существует. Пропуск '$file'.\n"
        fi
    fi
done
```

9. ?????????? ??????????? ?? ???????????

Вместо рискованного авто-обновления, этот скрипт проверяет наличие доступных обновлений и отправляет их список администратору на почту. Это позволяет принять взвешенное решение об обновлении вручную.

```
#!/bin/bash
ADMIN_EMAIL="admin@example.com"
#Обновляем список пакетов в тихом режиме
apt-get update > /dev/null
#Получаем список пакетов, доступных для обновления
UPGRADABLE=$(apt-get -s upgrade | awk '/^Inst/ {print $2}')
if [ -n "$UPGRADABLE" ]; then
    (echo "На сервере $(hostname) доступны следующие обновления:"; echo; echo "$UPGRADABLE") |
mail -s "System Updates Available" "$ADMIN_EMAIL"
fi
```

Revision #3
Created 5 October 2025 12:04:42 by Admin
Updated 5 October 2025 12:24:08 by Admin