

1. Создайте скрипт на языке bash с именем template_task.sh, делающий следующее:

- При запуске проверяет, что имя скрипта не совпадает с template_task.sh, если совпадает - выходит с уведомлением «я бригадир, сам не работаю»
- При запуске дописывает в файл report_имя_скрипта_без_полного_пути.log в рабочем каталоге информацию: [PID] ДАТА ВРЕМЯ Скрипт запущен
- Генерирует случайное число от 30 до 1800 и ждет такое количество секунд
- Дописывает в файл report_имя_скрипта_без_полного_пути.log сообщение: [PID] ДАТА ВРЕМЯ Скрипт завершился, работал N минут

vi template_task.sh

```
root@eltex-practice2-pg2-v14:~# cat template_task.sh
#!/bin/bash

# Проверка имени скрипта
script_name=$(basename "$0")
if [ "$script_name" = "template_task.sh" ]; then
    echo "я бригадир, сам не работаю"
    exit 1
fi

# Создание имени лог-файла
log_file="report_${script_name%.*log}"

# Запись начала работы
echo "[$$] $(date '+%Y-%m-%d %H:%M:%S') Скрипт запущен" >> "$log_file"

# Генерация случайного числа от 30 до 1800
sleep_time=$(( RANDOM % 1771 + 30 ))

# Ожидание
sleep "$sleep_time"

# Расчет минут работы
minutes=$(( sleep_time / 60 ))

# Запись завершения работы
echo "[$$] $(date '+%Y-%m-%d %H:%M:%S') Скрипт завершился, работал $minutes минут" >> "$log_file"
root@eltex-practice2-pg2-v14:~#
```

```
root@eltex-practice2-pg2-v14:~# chmod +x template_task.sh
```

```
root@eltex-practice2-pg2-v14:~# ./template_task.sh
я бригадир, сам не работаю
```

Но работает работник – worker.sh

```
root@eltex-practice2-pg2-v14:~# cp template_task.sh worker.sh
root@eltex-practice2-pg2-v14:~# ./worker.sh
root@eltex-practice2-pg2-v14:~#
```

```
root@eltex-practice2-pg2-v14:~# cat report_worker.log
[43597] 2025-10-18 05:12:15 Скрипт запущен
[43597] 2025-10-18 05:28:04 Скрипт завершился, работал 15 минут
```

2. Создайте скрипт на языке bash с именем observer.sh, читающий файл конфигурации со списком скриптов observer.conf, проверяющим их наличие в списке работающих процессов поиском в /proc и запускающих их в отключенном от терминала режиме (nohup) в случае отсутствия в нем. Информация о перезапуске дописывайте в файл observer.log

```
root@eltex-practice2-pg2-v14:~# vi observer.sh
root@eltex-practice2-pg2-v14:~# cat observer.sh
#!/bin/bash

config_file="observer.conf"
log_file="observer.log"

# Проверяем наличие файла конфигурации
if [ ! -f "$config_file" ]; then
    echo "$(date): Файл конфигурации $config_file не найден" >> "$log_file"
    exit 1
fi

# Читаем файл конфигурации построчно
while IFS= read -r script; do
    # Пропускаем пустые строки и комментарии
    [[ -z "$script" || "$script" =~ ^# ]] && continue

    # Проверяем наличие процесса
    if ! pgrep -f "$script" > /dev/null; then
        echo "$(date): Скрипт $script не найден, запускаем..." >> "$log_file"
        nohup bash "$script" > /dev/null 2>&1 &
        echo "$(date): Скрипт $script запущен с PID $!" >> "$log_file"
    fi
done < "$config_file"
```

3. Настройте запуск observer.sh посредством cron по расписанию – 1 раз в минуту

```
root@eltex-practice2-pg2-v14:~# realpath observer.sh
/root/observer.sh
root@eltex-practice2-pg2-v14:~# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]: 2
crontab: installing new crontab
```

```

* * * * /root/observer.sh
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)

```

* * * * = каждую минуту, каждый час, каждый день

4. Создайте несколько символьных ссылок на файл `template_task.sh` с различными именами (рабочие задачи), добавьте в файл конфигурации `observer.conf` соответствующие записи об этих задачах, включая исходный файл `template_task.sh`

```

root@eltex-practice2-pg2-v14:~# ln -s template_task.sh task1.sh
root@eltex-practice2-pg2-v14:~# ln -s template_task.sh task2.sh
root@eltex-practice2-pg2-v14:~# ln -s template_task.sh task3.sh
root@eltex-practice2-pg2-v14:~# echo "task1.sh" >> observer.conf
root@eltex-practice2-pg2-v14:~# echo "task2.sh" >> observer.conf
root@eltex-practice2-pg2-v14:~# echo "task3.sh" >> observer.conf

```

```

root@eltex-practice2-pg2-v14:~# ls -la task*.sh
lrwxrwxrwx 1 root root 16 Oct 18 06:24 task1.sh -> template_task.sh
lrwxrwxrwx 1 root root 16 Oct 18 06:24 task2.sh -> template_task.sh
lrwxrwxrwx 1 root root 16 Oct 18 06:24 task3.sh -> template_task.sh
root@eltex-practice2-pg2-v14:~# cat observer.conf
task1.sh
task2.sh
task3.sh

```

5. Соберите статистику работы в виде набора файлов `report_*.log`, `observer.log`, приложите их вместе с исходными текстами скриптов в качестве отчета в виде сжатого архива `tar`. Не забудьте остановить процесс, удалив задачу в `cron`!

```
root@eltex-practice2-pg2-v14:~# tar -czf practice3_report.tar.gz *.sh observer.conf report_ta
sk*.log
root@eltex-practice2-pg2-v14:~# tar -tzf practice3_report.tar.gz
observer.sh
task1.sh
task2.sh
task3.sh
template_task.sh
observer.conf
report_task1.log
report_task2.log
report_task3.log
root@eltex-practice2-pg2-v14:~#
```

```
root@eltex-practice2-pg2-v14:~# crontab -r
root@eltex-practice2-pg2-v14:~# crontab -l
no crontab for root
root@eltex-practice2-pg2-v14:~#
```