

Unix Final

Team19 108072244 邱煒甯 108032053 陳凱揚 112065525 簡佩如

Usage

```
make

./q1 &
kill -USR1 %1
# Output: T3 handling SIGUSR1
kill -INT %1
# Output: T1 handling SIGINT
kill -TERM %1
# Output: T2 handling SIGTERM

./q2 1000000
# Output: Sleep time: 1000438 us

./q3
# Output: Alarm!
```

Q1

We first set up a signal set `sigset_t set`, then use `sigaddset` to add `SIGINT`, `SIGTERM` and `SIGUSR1` to this set, and then use `pthread_sigmask` to block these signals. In `pthread_create`, we use `handle_sigint` to process the `SIGINT` signal. First create a signal collection and add the `SIGINT` signal to this collection. Then use the `pthread_sigmask` function to block these signals. Finally, use the `sigwait` function to wait for the `SIGINT` signal. Once received, it will print `"T3 handling SIGUSR1\n"`. We use the same method to let `handle_sigterm` and `handle_sigusr1` handle the `SIGTERM` and `SIGUSR1` signals respectively. Finally, `pthread_join` waits for these threads to complete. This means that the main thread will wait for all child threads to finish.

Q2

Define two `timeval` structures `start` and `end` for calculating sleep time. To obtain the current time before sleeping, use the `sleep_us` function to make the program sleep for the specified number of microseconds, and obtain the current time again after sleeping. Calculate the duration of sleep and output it in microseconds.

Q3

We set up a `sigaction` structure to specify how to handle the `SIGALRM` signal when it occurs. We set `sa_handler` to the `alarm_handler` function and `sa_flags` to `SA_RESTART`. Then we call the `sigaction` function to set the handler for the `SIGALRM` signal. The handler will print `"Alarm!\n"`.