



School of Computer Science

COMP30640

Lab 6
Synchronisation II

Teaching Assistant:	Thomas Laurent
Coordinator:	Anthony Ventresque
Date:	Friday 19 th October, 2018
Total Number of Pages:	2

1 Atomic Operations in Bash.

Semaphores are based on the concept of atomic operations - i.e., operations that cannot be split in multiple instructions by the system and cannot be interrupted while they're executed. In particular (see lecture 7) we need an atomic operation that can perform a **test** (if) and **modify/write a value**. There are a few atomic operations in Bash that we could use, as they perform something (write/modify) and give an exit value that we can use in a conditional structure. One of them is `ln`, which creates a link between two files.

1. Create a file (or use one pre-existing file, for instance one of your previous scripts, like `hello.sh`)

2. Create a link using the following syntax:

```
$> ln hello.sh link.sh
```

3. Open the new file you've just created (`link.sh` in my example). What do you notice?
4. run the command `ls -l`. What difference(s) do you see between the target file and the linked (source) file?

5. Run the same command again, with a different link name, for example:

```
$> ln hello.sh link2.sh
```

6. What is the exit value?
7. Run the exact same command again. What is the exit value now?

2 Semaphores.

Now that we have an atomic operation and that we know what its exit code is, we can use it to implement our semaphores. In particular, our P and V can be used with an **existing file** given as argument. P would then try to create a link with `ln` and if successful the critical section would be available or the process would have to wait.

Create two scripts:

P.sh containing the following code:

```
#!/bin/bash

if [ -z "$1" ]; then
    echo "Usage $0 mutex-name"
    exit 1
elif [ ! -e "$1" ]; then
    echo "Target for the lock must exist"
    exit 2
else
    while ! ln "$1" "$1-lock" 2>/dev/null; do
        sleep 1
    done
    exit 0
fi
```

And V.sh containing this code:

```
#!/bin/bash

if [ -z "$1" ]; then
    echo "Usage $1 mutex-name"
    exit 1
else
    rm "$1-lock"
    exit 0
fi
```

8. Run ./P.sh with one parameter: an **existing file**. For instance create a new file sema, using the command `touch sema`, and execute the following command: `./P.sh sema`. What happens? (look at the files in your directory)
9. Run ./P.sh with the same parameter. What happens? Why? (look at the files in your directory)
10. kill ./P.sh with the sequence control + c. Run ./V.sh. What happened? (look at the files in your directory)

Example of Semaphores in Use.

11. Revisit the last section (section 6, questions 1 to 3) of last week's practical. Do you see the problem? Run the following command to help you:

```
$> ./run_write.sh; wc -l f*
```
12. Identify the critical section in `write.sh` and add calls to `./P.sh` and `./V.sh` (with the correct argument) to ensure a mutual exclusion.
Hint: You will also need to change P.sh from the first example *very* slightly - look at the first \$1 in the while loop.
13. Did you synchronise the scripts?