Assignment2 supplement sheet

Q1: `ssh $robot_user@$robot -t "bash -i -c 'roscd && cd .. && catkin build'"` under `develop.sh`

Error:

```
Traceback (most recent call last):
  File "/usr/bin/catkin", line 33, in <module>
    sys.exit(load_entry_point('catkin-tools==0.9.0', 'console_scripts', 'catkin')())
  File "/usr/lib/python3/dist-packages/catkin_tools/commands/catkin.py", line
    catkin_main(sysargs)
  File "/usr/lib/python3/dist-packages/catkin_tools/commands/catkin.py", line
    sys.exit(args.main(args) or 0)
  File "/usr/lib/python3/dist-packages/catkin_tools/verbs/catkin_build/cli.py", l
    mark_space_as_built_by(ctx.build_space_abs, 'catkin build')
  File "/usr/lib/python3/dist-packages/catkin_pkg/tool_detection.py", line 78, ir
    with open(marker_path, 'w') as f:
PermissionError: [Errno 13] Permission denied: '/home/demo/ws_ros/build/.bu
```

Ans: just reopen the terminal, it fixed

Q2: What is SSH? How it would work for the TAMS computer?

Ans: The SSH is a safe protocol for you to connect remotely to the computer you want, for any of the computer that allow the SSH connection you can remotely access it. For example, in the lab room the computer IP `134.100.13.191` is also named as `tams191` so you can directly connection through all the computer in the campus directly `ssh yourname@tasm191` , and it would ask about you your password to login. If you are outside the campus, you would need to ssh to the campus VPN computer first like `ssh yourname@` `rzssh1.informatik.uni-hamburg.de` and then from this computer you can connect to any of tams computer. The computer we offer for remote login in the computer pool is from tams92-tams98. Sometimes one do not work, then try another one.

Reference: https://www.inf.uni-hamburg.de/inst/irz/it-services/private-devices/ssh-service.html

Q3: If I do not want to input password every time through using SSH?

Ans: You can create a SSH-key for it, check the guidance on website Arbeiten mit SSH-Schlüsselpaaren : RRZ : Universität Hamburg. First is `ssh-keygen` command so you create the private key and public key store in `~/.ssh/` From there you can share the public key to which computer you would like to log in in the folder `remote computer: ~/.ssh/know_hosts` , so that every time you log in it would use your own private key to match the public key in the remote computer so it will pass the login credential so you do not need to enter any password anymore.