# Assignment 1

## Part 1:

### 1.

I would use monolithic because there is no sending stuff between modules making it faster.

### 2.

It is slower because it has to copy the data each time. I would suggest combining commonly used modules to make a hybrid kernel like windows.

### 3.

The child processes become orphans and the parent gets changed to the init process.

### 4.

If the process runs into a critical error it might crash. I the user sends an interrupt to close a program, it would stop the process. If the deadline is up after a process has been hanging for too long, it will also close.

### 5.

I'm pretty sure that is already the case, but if short term was even faster and more efficient and long term was more slow and inefficient, small quick processes would be cleared up instantly, while longer processes would most likely build up overtime and lead to them not starting.

## Part 2:

I don't have a windows computer so I used a shell script.

'clear' clears the terminal window.
'uname -r' prints "6.11.1-arch-1" which I think is the operating system and version.
'read -p "Press any key to continue..."' waits for user input.
'ls /usr/bin' list the files in the directory.