

## Problem Sheet #9

### Problem 9.1: */proc/hangman in the linux kernel*

(2+4+2+2 = 10 points)

The game hangman is a word guessing game. The player has to guess a word (or phrase) by guessing the characters that make up the word (or phrase). Your task is to implement a Linux kernel module implementing the hangman game. The interface to the hangman game is the proc filesystem. Lets simply look at an example execution of the game:

```
$ sudo insmod ./hangman.ko word="'Hello World!'" duration=60
$ cat /proc/hangman/e
_ e _ _ _ _ _ _ _ _ ! (guess 1: 'e', time left: 60 seconds)
$ cat /proc/hangman/l
_ e l l _ _ _ _ l _ _ ! (guess 2: 'l', time left: 56 seconds)
$ cat /proc/hangman/a
_ e l l _ _ _ _ l _ _ ! (guess 3: 'a', time left: 45 seconds)
$ cat /proc/hangman/d
_ e l l _ _ _ _ l d _ ! (guess 4: 'd', time left: 38 seconds)
$ cat /proc/hangman/h
H e l l _ _ _ _ l d _ ! (guess 5: 'h', time left: 35 seconds)
$ cat /proc/hangman/o
H e l l o _ o _ l d _ ! (guess 6: 'o', time left: 33 seconds)
$ cat /proc/hangman/w
H e l l o   W o _ l d _ ! (guess 7: 'w', time left: 28 seconds)
$ cat /proc/hangman/r
H e l l o   W o r l d _ ! (guess 8: 'r', solved with 8 guesses)
$ cat /proc/hangman/x
H e l l o   W o r l d _ ! (guess 9: 'x', solved with 8 guesses)
$ sudo rmmmod hangman
```

There are, of course, ways to play the game using brute-force:

```
$ sudo insmod ./hangman.ko word="'Hacking Linux!'"
$ cat /proc/hangman/*
_ a _ _ _ _ _ _ _ _ _ _ ! (guess 1: 'a', time left: 60 seconds)
_ a _ _ _ _ _ _ _ _ _ _ ! (guess 2: 'b', time left: 60 seconds)
_ a c _ _ _ _ _ _ _ _ _ _ ! (guess 3: 'c', time left: 60 seconds)
_ a c _ _ _ _ _ _ _ _ _ _ ! (guess 4: 'd', time left: 60 seconds)
_ a c _ _ _ _ _ _ _ _ _ _ ! (guess 5: 'e', time left: 60 seconds)
_ a c _ _ _ _ _ _ _ _ _ _ ! (guess 6: 'f', time left: 60 seconds)
_ a c _ _ _ _ g _ _ _ _ _ _ ! (guess 7: 'g', time left: 60 seconds)
H a c _ _ _ _ g _ _ _ _ _ _ ! (guess 8: 'h', time left: 60 seconds)
H a c _ i _ g _ _ _ _ _ _ ! (guess 9: 'i', time left: 60 seconds)
H a c _ i _ g _ _ _ _ _ _ ! (guess 10: 'j', time left: 60 seconds)
H a c k i _ g _ _ _ _ _ _ ! (guess 11: 'k', time left: 60 seconds)
H a c k i _ g L i _ _ _ _ _ ! (guess 12: 'l', time left: 60 seconds)
H a c k i _ g L i _ _ _ _ _ ! (guess 13: 'm', time left: 60 seconds)
H a c k i n g L i n _ _ _ _ ! (guess 14: 'n', time left: 60 seconds)
H a c k i n g L i n _ _ _ _ ! (guess 15: 'o', time left: 60 seconds)
H a c k i n g L i n _ _ _ _ ! (guess 16: 'p', time left: 60 seconds)
H a c k i n g L i n _ _ _ ! (guess 17: 'q', time left: 59 seconds)
H a c k i n g L i n _ _ _ ! (guess 18: 'r', time left: 59 seconds)
H a c k i n g L i n _ _ _ ! (guess 19: 's', time left: 59 seconds)
H a c k i n g L i n _ _ _ ! (guess 20: 't', time left: 59 seconds)
H a c k i n g L i n u _ _ _ ! (guess 21: 'u', time left: 59 seconds)
```

```
H a c k i n g   L i n u _ !   (guess 22: 'v', time left: 59 seconds)
H a c k i n g   L i n u _ !   (guess 23: 'w', time left: 59 seconds)
H a c k i n g   L i n u x !   (guess 24: 'x', solved with 24 guesses)
H a c k i n g   L i n u x !   (guess 25: 'y', solved with 24 guesses)
H a c k i n g   L i n u x !   (guess 26: 'z', solved with 24 guesses)
```

The `word` parameter of the kernel module defines the word or phrase to guess. The `duration` module parameter defines the duration (in seconds) of the game; the game resets itself once the game duration has expired.

Characters are guessed by reading a file in `/proc/hangman` that is named by the guessed character. The content of the file shows the current status of the game.

Only alphabetical characters are guessed, punctuation characters or numbers appearing in a word or phrase are not guessed. Note that characters are always guessed in lowercase and that a guessed character matches both lowercase and uppercase characters in the word or phrase.

The assignment can be broken down into the following steps:

- a) Implement a kernel module that creates a directory `/proc/hangman` and within it 26 files named after the characters 'a' to 'z'.
- b) Implement a data structure that keeps track of the characters already guessed and the count of guesses made. When a file in `/proc/hangman` is read, mark the character obtained from the file name as guessed and subsequently generate file content that indicates the status of the game play.
- c) Implement module parameters that enable to define the word to be guessed and the duration of the game while loading your kernel module.
- d) Add the automatic reset mechanism by setting a timeout for the game when the game started, i.e., when the first guess is made. Upon subsequent guesses, check whether the game has timed out or not. If the game has timed out, reset it to its initial state.

Make sure your code is robust. It should do something sensible if, for example, the word to guess is a zero-length string or does not contain any alphabetic characters.