

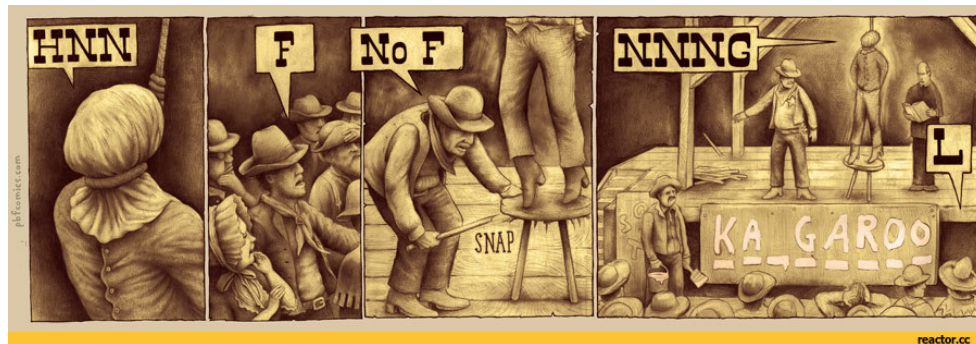
# Problem Set 3: Hangman

## Ciele

1. Vytvoriť vlastné funkcie podľa špecifikácie.
2. Rutinne pracovať s jednorozmerným poľom a refazcami.
3. Používať pole ako parameter (vstupný a výstupný) vo funkciách.

## Hangman

Určite ste už niekedy hrali hru so slovami Hangman, v ktorej je potrebné hádať slovo po písmenách. Uhádnuté písmená sa postupne dopisujú do tajničky a pri každom neúspešnom pokuse sa naopak nakreslí časť šibenice. Hra sa končí, keď hráč uhádne slovo alebo keď druhý hráč nakreslí celú šibenicu.



Obr. 1: Spoločenské prevedenie hry Hangman. Jej napodobovanie sa však neodporúča.

V našej implementácii bude prvým hráčom používateľ, ktorý bude hádať slovo tak, že vždy napíše na obrazovku jedno písmeno. Na uhádnutie slova bude mať **8 pokusov**. Druhým hráčom bude vždy počítač, ktorý si najskôr náhodne vyberie tajné slovo zo slovníka. Výber bude zabezpečený volaním funkcie `int get_word(char secret[])`, ktorá vráti slovo s max. dĺžkou **15** znakov (túto funkciu už budete mať pripravenú).

Vašou úlohou bude naprogramovať štyri funkcie:

- `int is_word_guessed(const char secret[], const char letters_guessed[])` - Zistí, či na základe hádaných písmen hráč uhádol tajné slovo (malé písmená).
- `void get_guessed_word(const char secret[], const char letters_guessed[], char guessed_word[])` - Aktualizuje hádané slovo (malé písmená) tak, že na tajných, zatiaľ neuhádnutých pozíciách bude obsahovať znak '\_' a na odkrytých, už uhádnutých miestach bude obsahovať konkrétne písmeno.
- `void get_available_letters(const char letters_guessed[], char available_letters[])` - Aktualizuje zoznam dostupných písmen, ktoré ešte neboli hádané (malé písmená).
- `void hangman(const char secret[])` - Obsahuje funkcionálnu samotnej hry.

Všetkých 5 funkcií (štyri, ktoré máte naprogramovať a funkcia `get_word()`, ktorú už dostanete) sa bude nachádzať v súbore `hangman.c`. Ich deklarácie sú uvedené v súbore `hangman.h`. Hlavný program sa nachádza v súbore `main.c`, ktorý načíta náhodné slovo pomocou funkcie `get_word()` a odovzdá ho funkcii `hangman()` na realizáciu samotnej hry.

Okrem požadovaných funkcií si môžete vytvoriť aj vlastné funkcie. Tieto však budú súkromné pre Váš modul (nebudú deklarované v hlavičkovom súbore `hangman.h`).

Na začiatku hry je hráč oboznámený s dĺžkou tajného slova. V každom kole je spracované práve jedno písmeno a hráč je okamžite oboznámený o tom, či sa písmeno nachádza v tajnom slove alebo nie. Po každom kole by sa tiež malo zobrazíť samotné slovo tak, aby bolo jasné, ktoré písmená už boli uhádnuté a ktoré zatiaľ nie.

Nezabudnite hráčov pripomenúť, koľko pokusov má ešte k dípózií. Hráč stráca pokus iba ak háda nesprávne, t.j. dané písmeno sa v tajnom slove nenachádza. Ak hráč zadá to isté písmeno viackrát, neprichádza o pokus, oznámte mu však, že sa jeho tip už zopakoval.

Hra končí, ak hráč uhádne tajné slovo alebo ak sa hráčovi minú všetky pokusy. Ak hráč neuhádne tajné slovo, toto slovo je vypísané na obrazovku.

## Úloha 1: Je slovo uhádnuté?

Naprogramujte funkciu `int is_word_guessed(const char secret[], const char letters_guessed[])` s dvoma parametrami:

- `const char secret[]` - Refazec reprezentujúci tajné slovo (malé písmená)
- `const char letters_guessed[]` - Refazec so všetkými doposiaľ hádanými písmenami

Funkcia vráti hodnotu **1**, ak tajné slovo uložené v parametri `secret` bolo uhádnuté (je ho možné zostaviť z hádaných písmen uložených v parametri `letters_guessed`). V opačnom prípade vráti funkcia hodnotu **0**.

Príklad použitia funkcie:

```
printf("%d\n", is_word_guessed("secret", "aeiou"));
// prints: 0
printf("%d\n", is_word_guessed("hello", "aeihoul"));
// prints: 1
```

**Poznámka:** Pre znaky v celom programe platí, že ide o malé písmená abecedy.

## Úloha 2: Aktualizácia hádaného slova

Naprogramujte funkciu `void get_guessed_word(const char secret[], const char letters_guessed[], char guessed_word[])` s tromi parametrami:

- `const char secret[]` - Refazec reprezentujúci tajné slovo (malé písmená)
- `const char letters_guessed[]` - Refazec so všetkými doposiaľ hádanými písmenami
- `char guessed_word[]` - Výstupný parameter funkcie, ktorý bude reprezentovať refazec hádaného slova

Funkcia nevráti žiadnu hodnotu.

Funkcia aktualizuje hádané slovo `guessed_word` tak, že na tajných, zatiaľ neuhádnutých pozíciách bude obsahovať znak '\_' a na odkrytých, už uhádnutých miestach bude obsahovať konkrétne písmeno.

Príklad použitia funkcie:

```
char result[30];
get_guessed_word("container", "arpstxgoieyu", result);
// result = "_o_tai_er"
```

**Poznámka:** Nezabudnite, že pre znaky v celom programe platí, že ide o malé písmená abecedy.

## Úloha 3: Aktualizácia dostupných písmen

Naprogramujte funkciu `void get_available_letters(const char letters_guessed[], char available_letters[])` s dvoma parametrami:

- `char letters_guessed[]` - Refazec so všetkými doposiaľ hádanými písmenami (malé písmená)
- `char available_letters[]` - Refazec abecedy okrem doposiaľ hádaných písmen. Výstupný parameter funkcie.

Funkcia nevráti žiadnu hodnotu.

Funkcia aktualizuje refazec dostupných písmen `available_letters` vynechaním už hádaných písmen pomocou parametra `letters_guessed`. Tento zoznam písmen bude usporiadaný vzostupne.

Príklad použitia funkcie:

```
char result[30];
get_available_letters("arpstxgoieyu", result);
// result = "bcdfhjklmnqvwz"
```

**Poznámka:** Nezabudnite, že pre znaky v celom programe platí, že ide o malé písmená abecedy.

## Úloha 4: Hra

Naprogramujte funkciu `void hangman(const char secret[])` s jedným parametrom:

- `const char secret[]` - Refazec reprezentujúci tajné slovo

Funkcia nevráti žiadnu hodnotu.

Funkcia realizuje interaktívnu hru medzi používateľom a počítačom s využitím Vami vytvorených funkcií `int is_word_guessed()`, `void get_guessed_word()` a `void get_available_letters()`.

Na začiatku hra oboznámi hráča s dĺžkou tajného slova. V každom kole hry sa spracuje práve jedno písmeno a hráč je oboznámený s počtom zostávajúcich pokusov a refazcom dostupných písmen. Po zadaní znaku je hráč okamžite informovaný o úspešnosti jeho tipu spolu s výpisom aktuálneho hádaného slova s vyznačenými uhádnutými písmenami.

Hráč však môže z klávesnice zadať ľubovoľné písmeno - aj veľké aj malé! V tomto prípade sú si písmená rovnocenné (nezáleží na ich veľkosti). Túto funkcionálnosť naprogramujte priamo vo funkcii `hangman()`.

Hra by mala po každom pokuse vypísať počet zostávajúcich pokusov. Počet pokusov sa znižuje o jedno, ak bol tip hráča nesprávny. Naopak, počet pokusov zostáva rovnaký, ak bol tip hráča správny, ale aj keď hráč zadá opakovane rovnaký tip. Vtedy je potrebné o

tom hráča informovať.

Slovo, ktoré ste získali ako výstup z funkcie `get_guessed_word()`, kvôli zvýšeniu čitateľnosti vypisujte v tvare `l a _ _ s _ a _ e`.

Ak hráč zadá celé slovo (nie iba 1 písmeno), hra ihneď končí a hráčovi oznámi, či vyhral alebo nie.

## Príklady použitia programu

Ak hráč uhádne tajné slovo, hra končí:

```
$ ./hangman
Welcome to the game, Hangman!
I am thinking of a word that is 7 letters long.
-----
You have 8 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter: l
Good guess: _ _ _ _ l _ _
-----
You have 8 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter: e
Good guess: _ _ _ _ l e _
-----
You have 8 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter: d
Oops! That letter is not in my word: _ _ _ _ l e _
-----
You have 7 guesses left.
Available letters: abcfghijklmnopqrstuvwxyz
Please guess a letter: a
Oops! That letter is not in my word: _ _ _ _ l e _
-----
You have 6 guesses left.
Available letters: bcfghijklmnopqrstuvwxyz
Please guess a letter: U
Good guess: _ u _ _ l e _
-----
You have 6 guesses left.
Available letters: bcfghijklmnopqrstuvwxyz
Please guess a letter: u
Oops! You've already guessed that letter: _ u _ _ l e _
-----
You have 6 guesses left.
Available letters: bcfghijklmnopqrstuvwxyz
Please guess a letter: s
Good guess: _ u _ _ l e s
-----
You have 6 guesses left.
Available letters: bcfghijklmnopqrstuvwxyz
Please guess a letter: p
Good guess: p u _ p l e s
-----
You have 6 guesses left.
Available letters: bcfghijklmnopqrstuvwxyz
Please guess a letter: r
Good guess: p u r p l e s
-----
Congratulations, you won!
```

Ak sa hráčovi minie všetkých 8 pokusov, hra končí spolu s vypísaním tajného slova:

```
$ ./hangman
Welcome to the game, Hangman!
I am thinking of a word that is 10 letters long.
-----
You have 8 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter: a
Oops! That letter is not in my word: _ _ _ _ _ _ _ _ _ _
-----
You have 7 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter: e
Good guess: _ _ _ e _ e _ _ _ _
-----
...
-----
You have 1 guesses left.
Available letters: hjklmnopqrstuvwxyz
Please guess a letter: h
```

Oops! That letter is not in my word: u \_ d e \_ e \_ \_ e d

-----

Sorry, you ran out of guesses. The word was undeserved.

Ak hráč zadá celé slovo, hra končí a hráč ihneď vie, či vyhral alebo nie:

\$ ./hangman

Welcome to the game, Hangman!

I am thinking of a word that is 4 letters long.

-----

You have 8 guesses left.

Available letters: abcdefghijklmnopqrstuvwxyz

Please guess a letter: **e**

Oops! That letter is not in my word: \_ \_ \_ \_

-----

You have 7 guesses left.

Available letters: abcd~~f~~ghijklmnopqrstuvwxyz

Please guess a letter: **ba**ll

Congratulations, you won!

\$ ./hangman

Welcome to the game, Hangman!

I am thinking of a word that is 4 letters long.

-----

You have 8 guesses left.

Available letters: abcdefghijklmnopqrstuvwxyz

Please guess a letter: **o**

Good guess: \_ o \_ \_

-----

You have 8 guesses left.

Available letters: abcdefghijklmnpqrstuvwxyz

Please guess a letter: **wo**rd

Sorry, bad guess. The word was goal.

Ak hráč zadá znak, ktorý nie je písmeno, počet zvyšných pokusov sa nezmení:

\$ ./hangman

Welcome to the game, Hangman!

I am thinking of a word that is 7 letters long.

-----

You have 8 guesses left.

Available letters: abcdefghijklmnopqrstuvwxyz

Please guess a letter: **a**

Good guess: \_ a \_ \_ a \_

-----

You have 8 guesses left.

Available letters: bcdefghijklmnopqrstuvwxyz

Please guess a letter: **e**

Oops! That letter is not in my word: \_ a \_ \_ \_ a \_

-----

You have 7 guesses left.

Available letters: bcd~~f~~ghijklmnopqrstuvwxyz

Please guess a letter: **@**

Oops! '@' is not a valid letter: \_ a \_ \_ \_ a \_

-----

You have 7 guesses left.

Available letters: bcd~~f~~ghijklmnopqrstuvwxyz

Please guess a letter: **h**

Good guess: h a \_ \_ \_ a \_

-----

You have 7 guesses left.

Available letters: bcd~~f~~gijklmnopqrstuvwxyz

Please guess a letter: **n**

Good guess: h a n \_ \_ a n

-----

You have 7 guesses left.

Available letters: bcd~~f~~gijklmopqrstuvwxyz

Please guess a letter: **hang**man

Congratulations, you won!

## Odovzdávanie projektu

Zadanie odovzdajte do **18.11.2022** (piatok). Posledné testovanie prebehne v tento deň o polnoci.

Zadanie sa odovzdáva prostredníctvom systému na správu verzií [Git](#) na serveri [git.kpi.fei.tuke.sk](#).

Názov Vášho projektu musí byť v tvare: **zap-2022-id**.

Projekt musí mať nasledujúcu štruktúru priečinkov a súborov:

```
.
├── ps3
│   ├── hangman.c
│   ├── hangman.h
│   └── main.c
└── README
```

Význam jednotlivých súborov je nasledovný:

- **README** - Súbor, v ktorom bude uvedená Vaša skupina, ktorú navštevujete na cvičeniach v tvare:

```
GROUP : C1
```

- `/ps3/hangman.c` - Zdrojový kód knižnice pre hru *hangman*.
- `/ps3/hangman.h` - Hlavičkový súbor knižnice *hangman*.
- `/ps3/main.c` - Zdrojový kód obsahujúci funkciu `main()`.

**Upozornenie:** Je dôležité, aby Vaše súbory zachovali uvedenú štruktúru. Ak sa niektorý zo súborov síce v repozitári nachádza, ale v inom priečinku, bude to považované za chybu a takýto projekt nebude považovaný za správny.

**Upozornenie:** Pri názvoch priečinkov, súborov a obsahu súboru **README záleží na veľkosti písmen!**

**Poznámka:** Ak sa vo Vašom projekte budú nachádzať ďalšie súbory okrem požadovaných, ich existencia **nebude** považovaná za chybu.

**Upozornenie:** Ak používate **Makefile**, preklad programu uskutočnite pomocou príkazu `make all`.

## Kostra projektu

Z nasledujúceho odkazu si stiahnite súbor [hangman.zip](#), ktorý obsahuje kostru projektu. Tento balíček obsahuje nasledujúce súbory:

- `words.txt` - Súbor, ktorý obsahuje 55900 rozličných anglických slov.
- `hangman.h` - Hlavičkový súbor, v ktorom sa nachádzajú deklarácie všetkých požadovaných funkcií.
- `hangman.c` - Zdrojový súbor, v ktorom sa bude nachádzať Vaša výsledná implementácia, a v ktorom sa nachádza už implementovaná funkcia `get_word()` pre načítanie náhodného slova zo súboru `words.txt`.
- `Makefile` - Súbor pre preklad programu, ktorý uskutočnite pomocou príkazu `make all`.

V prostredí *OS Linux* môžete pre stiahnutie použiť príkaz `wget` v tvare:

```
wget http://kurzy.kpi.fei.tuke.sk/zap/resources/hangman.zip
```

## Hodnotenie a testovanie

Za zadanie môžete získať **max. 10 bodov**. Lubovoľné 2 body sa započítajú do zápočtu, ostatné body sa započítajú do skúšky. Počet získaných bodov sa bude odrážať od výsledku testov, ktorými Vaše zadanie úspešne prejde. Overovať sa bude:

- Štruktúra Vášho projektu (či sa v ňom nachádzajú všetky potrebné súbory).
- Funkčnosť Vašej implementácie.

Váš kód sa bude prekladať prekladačom `gcc` s nasledovnými prepínačmi:

```
gcc -std=c11 -Werror -Wall -lm
```

Ak používate nami poskytnutý **Makefile**, preklad programu uskutočnite pomocou príkazu:

```
make all
```

Za chybu sa bude považovať:

- Ak vo Vašej implementácii použijete **globálnu** premennú.
- Ak počas prekladu dôjde ku chybe (upozornenia sú priamo konvertované na chyby).
- Ak Vaša implementácia neprejde niektorým z testov.

Testovanie Vašich riešení sa bude vykonávať automaticky každé 3 hodiny a to konkrétne o 0300, 0600, 0900, 1200, 1500, 1800, 2100 a 2400.

Vaše riešenia prejdú kontrolou originality. Preto sa pri práci na Vašom zadaní správajte podľa pravidiel [etického kódexu](#)! V prípade, že odovzdáte zadanie, ktoré nie je Vaše, môžete byť vylúčený z predmetu!

© Copyright 2022 [KPI FEI TUKE](#).

Poháňané CSS rámcem [Bootstrap v4](#)