

MUNI
FI

Cvičenie 06

Dynamická alokácia

Dynamická alokácia

- Používaná, ak dopredu nevieme, koľko miesta budeme potrebovať
- Možné alokovať ľubovoľné množstvo bajtov, nikto nesľubuje, že ich dostaneme
- Alokovaný priestor v pamäti je na halde
- Spôsobuje haldu problémov
- Všetky operácie musí vykonať programátor (program) explicitne
 - Alokácia
 - Uvoľnenie = dealokácia
 - Navariť, vyprať, vyžehliť
 - Prosto ako intrák

Funkcie pre alokáciu - potrebujem

- Z knižnice `stdlib.h`
- `void* malloc(size_t n)`
 - Skúsi alokovať `n` bajtov
 - Úspech → ukazovateľ na začiatok alokovaného priestoru
 - Neúspech → `NULL` ukazovateľ
 - Alokované bajty nie sú inicializované
- `void* calloc(size_t n, size_t sizeItem)`
 - Alokuje `n * sizeItem` bajtov
 - Inicializuje alokovanú pamäť na `0`

Funkcia pre realokáciu – potrebujem viac

– Z knižnice `stdlib.h`

– `void* realloc(void* ptr, size_t size)`

– `ptr` – Ukazovateľ na začiatok skôr alokovanej pamäte

– `ptr == NULL` → rovnaké ako `malloc`

– `ptr != NULL` → skúsi alokovať `size` bajtov

Neúspech → vráti `NULL` ukazovateľ, pôvodný `ptr` nie je uvoľnený

Úspech → vráti ukazovateľ na začiatok alokovaného priestoru, ale nemusí byť rovnaký ako `ptr`, pôvodná pamäť je uvoľnená

`size` > pôvodná veľkosť → môže presunúť na nové miesto, nový priestor neinicializovaný

`size` < pôvodná veľkosť → iba skrúti

`size` = 0 → možno ako `free`, možno spácha zločin ☐☐

Funkcia pre dealokáciu – už nepotrebujem

- Z knižnice `stdlib.h`
- `void free(void* ptr)`
 - `ptr` – Ukazovateľ na začiatok skôr alokovanej pamäte
 - Nemaže obsah pamäte
 - Vhodné po volaní nastaviť `ptr` na `NULL`

Funkcia pre nastavenie pamäte na hodnotu

- Z knižnice `stdlib.h`
- `void* memset(void* ptr, int value, size_t num)`
 - Nastaví `num` bajtov začínajúc na `ptr` na `value`
 - `memset` > `for` + `=`

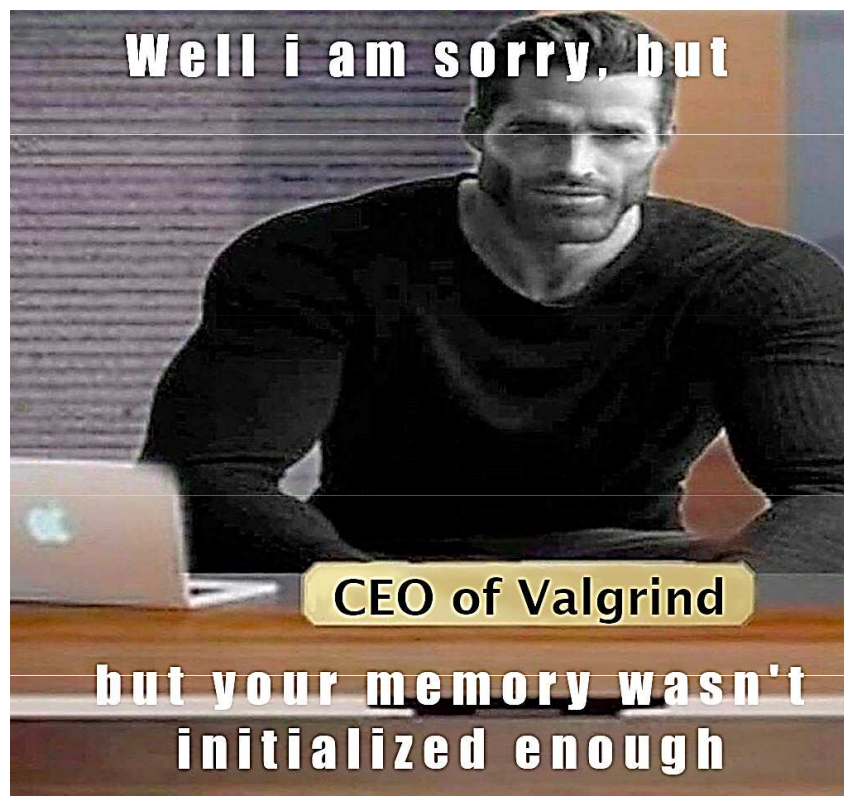
Ukážka

Memory leak

- Miesto v pamäti bolo (úspešne) naalokované
- Ukazovateľ na dané miesto bol stratený – prepísaný alebo ani neuložený
- Nemáme možnosť uvoľniť naalokovanú pamäť
- Môžeme kupovať novú RAMku □ □

- Jedine že by...
...sme použili Valgrind

Valgrind



Valgrind Memcheck

- Skompilovať s prepínačom `-g` → Valgrind ukáže napr. názvy funkcií, v ktorých nastal problém
- `valgrind -v`
 - `--leak-check=full`
 - `--show-reachable=yes`
 - `[--track-origins=yes]`
 - `[--track-fds=yes]`
 - `./program arg1 arg2`

**Čo? Prečo? Ako?
???**