

Semestrální práce

Program: Test na prvočíslo

Zadání:

„Zadá se číslo uint16, program otestuje, zda se jedná o prvočíslo.“

Návrh řešení:

Nejdřív jsem si daný program udělal v programovacím jazyce Java:

```
public static void main(String[] args) {  
  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Zadejte cislo uint16: ");  
    int cislo = Integer.parseInt(sc.nextLine());  
    if(jePrvocislo(cislo)) {  
        System.out.println("Cislo " + cislo + " je prvocislo");  
    } else {  
        System.out.println("Cislo " + cislo + " není prvocislo");  
    }  
}  
  
public static boolean jePrvocislo(int cislo) {  
    if(cislo < 2) return false;  
    if(cislo == 2) return true;  
    for(int i = 3; i < cislo; i++) {  
        if(cislo % i == 0) return false;  
    }  
    return true;  
}
```

Takto jsem chtěl také strukturovat program v HEW. Využívám obecných znalostí prvočísel a to:

- Čísla menší než 2 (tedy 1 a 0) nejsou prvočísla
- Číslo 2 je prvočíslem
- Všechna prvočísla větší než 2 jsou lichá
- Prvočíslo má pouze dva dělitele, jedničku a sebe

Řešení v HEW:

1. Program začíná na `_start`, kde se definuje zásobník a volá se systémová funkce **@syscall** s kódem **PUTS, 0x114** a odkazem na parametrický blok **#par_prompt**. Tím se vypíše do konzole zpráva s vybitnutím uživatele o vstup.
2. Dále se znovu volá systémová funkce, tentokrát s kódem **GETS, 0x113** a odkazem na blok **#par_vstup**. To načte od uživatele vstup a ten se uloží do registru **ER6**.
3. Program skočí do podprogramu **@prevod**, kde dochází k převodu vstupu uživatele v decimální soustavě na hodnotu v soustavě hexadecimální, ve které se provádí výpočty. Tato hodnota se uloží do registru **R0**. Podprogram končí.

4. Dále program skočí do podprogramu **@testPrime**, kde se provádí test na prvočíslo.
 - a. Do registru **R5** se přesune číslo 2 a porovná se se vstupem (**E0**). Pokud je vstup menší, tak se jedná o 1 nebo 0. Provede se podmíněný skok do **lab3**.
 - b. Dále se kontroluje zda vstup není roven číslu 2 (**R5**). Pokud ano, provede se podmíněný skok do **lab2**.
 - c. Následně kontrolujeme sudost čísla. Kopíruje se vstup (**E0**) do registru **ER4** a tam také probíhá dělení číslem dva (**R5**). Zbytek po dělení se nachází v **E4** a to porovnáme s nulou. Pokud zbytek se rovná nule tak je číslo sudé a určitě není prvočíslem. Číslo dva jsme kontrolovali už v předchozím kroku, proto nám to nedělá problém. Podmíněný skok do **lab3**.
 - d. Pokračujeme testem dělením. K číslu 2 (v **R5**) přičteme číslo jedna. Dostáváme 3. Porovnááme vstup s číslem v **R5** (momentálně rovno 3). Provádí se test dělením jako v bodě **c**). Pokud se zbytek rovná nule pak se provede podmíněný skok do **lab3**. Jinak cyklus pokračuje dál, nejdříve inkrementací „dělitele“ o hodnotu 2, vynulováním registru **ER4**, kde se provádělo dělení a posledně skokem na **@loop**. Tam se opět kontroluje zda vstup není roven číslu v **R5** (teď už 5) a program jede tak dlouho dokud se někdy dělitel nebude rovnat nule a skočí do **lab3** nebo dokud nedoběhne cyklus do konce, tím že se vstup rovná číslu v **R5** (jakési počítadlo a zároveň dělitel) a v tom případě se provede skok do **lab2**.
 - e. Při skoku do **lab2** se volá funkce **PUTS** s odkazem na parametrický blok **#par_isPrime**. Uživateli se objeví zpráva, že se jedná o prvočíslo, program skočí do **lab1**, kde dojde ke konci.
 - f. Při skoku do **lab3** se použije odkaz **#par_notPrime**. Uživateli se objeví zpráva, že vstup není prvočíslo a program se ukončí skokem do **lab1**.

Tabulka definovaných proměnných:

BP	Address	Name	BP	Address	Name
	000000	rs		0004B6	_erodata
	000113	GETS		01FF00	syscall
	000114	PUTS		FF4000	vstup
	000400	testPrime		FF4000	_data
	000400	.L0[]		FF4006	prompt
	000400	.L0[]		FF401D	is_prime
	00041E	loop		FF4038	not_prime
	000436	lab2		FF4058	par_vstup
	000448	lab3		FF405C	par_prompt
	00045A	_start		FF4060	par_isPrime
	00048A	lab1		FF4064	par_notPrime
	00048E	prevod		FF40CC	stack
	00049E	lab5		FF40CC	_ebss
	0004B0	lab6		FF40CC	_bss
	0004B6	.L0[]		FF40CC	_edata
	0004B6	__DTOR_END__		FF40CC	_end
	0004B6	__CTOR_LIST__		FFC000	_stack
	0004B6	etext			

Hodnoty registrů (v podprogramu):

Name	Value	R...
ER0	00000003	Hex
ER1	00FF4058	Hex
ER2	00000000	Hex
ER3	00000000	Hex
ER4	00010001	Hex
ER5	00000003	Hex
ER6	00FF4001	Hex
ER7	00FF40C8	Hex
PC	000400	Hex
CCR	10000000	I0---... Bin
EXR	01111111	-----... Bin
MACH	00000000	Hex
MACL	00000000	Hex

Zásobník (v podprogramu):

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	^
FF4060	00	FF	40	1D	00	FF	40	38	00	00	00	00	00	00	00	00	
FF4070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
FF4080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
FF4090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
FF40A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
FF40B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
FF40C0	00	00	00	00	00	FF	40	58	00	00	04	8A	00	00	00	00	
FF40D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00			▼