



Popis jazyka run

Jak promlouvatí v runách magických

Runový program se skládá z posloupnosti run a oddělovačů (při zápisu se značí buď mezerou nebo svislicí). Oddělovače dělí runy na jednotlivé *instrukce*. Čtení *instrukcí* probíhá takto:

- První přečtená runa určuje *příkaz*.
- Ostatní runy v instrukci jsou *parametrem* tohoto příkazu.

Příklad: Program „ $\mathfrak{W}\Psi\mathfrak{F}|\mathfrak{Y}\Delta\mathfrak{R}|\Psi$ “ má tři instrukce: První je tvořena příkazem \mathfrak{W} s parametrem $\Psi\mathfrak{F}$, druhá příkazem \mathfrak{Y} s parametrem $\Delta\mathfrak{R}$ a třetí příkazem Ψ s prázdným parametrem.

Parametry

Každá posloupnost run má svoji číselnou hodnotu, která je společně s přepisem run do latinky uvedena níže. Každé runa má přiřazené číslo od nuly do dvou, tedy posloupnost run je vlastně číslo ve trojkové soustavě.

Při zpracování *parametru* příkazu se tento parametr „rozbálí“. To znamená:

- Prázdný parametr má výchozí hodnotu 42.
- Parametr nezačínající runou Δ se rozbálí na svou číselnou hodnotu. Parametr prvního příkazu ukázkového programu je tedy 22 v trojkové soustavě, což je 8 v desítkové soustavě.
- Pokud parametr začíná runou Δ , pak se nejdříve rozbálí zbytek *parametru* za Δ a výsledkem je pak hodnota uložená v paměti na místě daném rozbalením zbytku *parametru*. V případě, že zbytek opět začíná Δ , tak se postupuje rekurzivně.

Příkladem je druhá instrukce ukázkového programu: Příkaz \mathfrak{Y} má parametr $\Delta\mathfrak{R}$, tedy dostane jako parametr hodnotu uloženou v paměti na adrese \mathfrak{R} , tedy na adrese 1.

Runa	Hodnota	Písmeno	Runa	Hodnota	Písmeno
\mathfrak{W}	0	a	\mathfrak{X}	0	g
\mathfrak{Q}	1	b	\mathfrak{R}	1	h
\mathfrak{P}	2	c	\mathfrak{M}	2	i
\mathfrak{H}	0	d	\mathfrak{Y}	0	j
\mathfrak{F}	1	e	\mathfrak{Z}	1	k
\mathfrak{I}	2	f	Δ	2	l

Celý ukázkový program lze tedy přepsat do latinky jako: acf jlh c.

Registry a pamatování

Pozor: Všechny hodnoty v následujících odstavcích jsou v desítkové soustavě, ačkoliv se runový počítač programuje soustavou trojkovou.

Runový počítač má jeden speciální *pracovní registr* a paměť obsahující 256 míst. *Pracovní registr* je namapovaný na desítkové adrese 42. Registr i každé místo v paměti jsou velké 8 bitů a mohou tedy obsahovat čísla 0 až 255. Na začátku je celá paměť i s registrem vynulovaná.

Pokud se program pokusí na libovolné místo zapsat hodnotu X větší než 255 nebo menší než 0, uloží místo toho hodnotu $X \bmod 256$. Uložení čísla -10 tedy místo toho uloží číslo 246, a uložení čísla 257 uloží místo toho číslo 1. Pokud se nějaká instrukce pokusí přistoupit na adresu A nacházející se mimo 0 až 255 (včetně), přistoupí místo toho na adresu $A \bmod 256$.

Ukázky programování runového

- Program „ $\mathfrak{W}\mathfrak{Q}\mathfrak{W}\mathfrak{F}|\mathfrak{M}\Delta$ “ (přepis: abae il) vypíše číslo 10.
- Program „ $\mathfrak{R}|\mathfrak{Y}|\mathfrak{M}\Delta$ “ (přepis: h j il) vypíše svůj vstup, ale jen tehdy, je-li různý od 0.
- Program „ $\mathfrak{M}\mathfrak{R}\mathfrak{F}|\mathfrak{X}\mathfrak{F}$ “ (přepis: ihf ge) donekonečna vypisuje číslo 5.
- Program „ $\mathfrak{R}\mathfrak{W}|\mathfrak{R}\mathfrak{Q}|\mathfrak{W}\Delta\mathfrak{W}|\mathfrak{Z}\Delta\mathfrak{Q}|\mathfrak{F}|\mathfrak{Y}|\mathfrak{X}|\mathfrak{Q}|\mathfrak{M}\Delta\mathfrak{W}|\mathfrak{X}|\Delta|\Delta\mathfrak{Q}|\mathfrak{M}\Delta\mathfrak{Q}|\Delta\Delta$ “ (přepis: ha hb ala dlb f j gb ila gl lb ilb ll) načte dvě čísla a vypíše z nich to větší.

Význam run

Zde je uvedený seznam jednotlivých run a jaké se ke které runě vážou příkazy pro runový počítač:

- ☉ \mathfrak{W} - Z vody povstane, v paměti nakrátko zůstane.
Do pracovního registru uloží svůj rozbalený parametr.
- ☉ \mathfrak{D} - Necht' Ti lid raků prodlouží paměť.
Zkopíruje hodnotu z pracovního registru na adresu, kterou dostane rozbalením svého parametru.
- ☉ \mathfrak{P} - Pán moří sečte Tvé hříchy i dobra.
Přičte k hodnotě v pracovním registru hodnotu získanou rozbalením parametru. Výsledek uloží zpět do pracovního registru.
- ☉ \mathfrak{H} - V podsvětí, Pluto protiklad Neptuniv jest.
Odečte od hodnoty v pracovním registru hodnotu získanou rozbalením parametru. Výsledek uloží zpět do pracovního registru.
- ☉ \mathfrak{F} - Na bitevním poli, jen ze dvou křížů stává se jeden, jinak zhynou oba!
Vezme hodnotu z pracovního registru a hodnotu získanou rozbalením parametru a provede s nimi bitový AND. Výsledek uloží zpět do pracovního registru.
- ☉ \mathfrak{G} - Pluto shovívavý jest, stačí mu kříž alespoň jeden, pak může projít dál.
Vezme hodnotu z pracovního registru a hodnotu získanou rozbalením parametru a provede s nimi bitový OR. Výsledek uloží zpět do pracovního registru.
 - **Speciální:** V případě, že nedostane žádný parametr, uloží do pracovního registru hodnotu 1, pokud při poslední instrukci \mathfrak{P} nebo \mathfrak{H} došlo k přetečení nebo podtečení. Pokud k přetečení nebo podtečení nedošlo, uloží hodnotu 0.
 - Pokud však ještě žádná z těchto instrukcí v programu neproběhla, zaplní se celá paměť desetinným rozvojem čísla π (tedy na pozici 0 bude číslo 3, na pozici 1 číslo 1, pak 4, 1, 5, 9, 2, ...).
- ☉ \mathfrak{Z} - Kozoroh, odchovaný Plutem, nedovolí ale jiný počet, než kříž právě jeden.
Vezme hodnotu z pracovního registru a hodnotu získanou rozbalením parametru a provede s nimi bitový XOR. Výsledek uloží zpět do pracovního registru.
- ☉ \mathfrak{A} - Na vršku pyramidy, jeho světlo do dále naviguje střelce.
 - Označí místo v programu tak, že se na něj \nearrow může odkázat.
 - Pokud runou \mathfrak{A} začíná nějaký parametr, pak je chování popsáno výše.
- ☉ \mathfrak{S} - Střelec mocným šípem přenese Tě do krajín vzdálených.
Označme jako P parametr tohoto příkazu:
 - Pokud někde v programu existuje instrukce $\mathfrak{A}P$, pak vykonávání programu pokračuje první takovou instrukcí.
 - Pokud ne, tak se P rozbálí, tím získáme číslo V . Pokud má program instrukci s číslem V , pak vykonávání programu pokračuje touto instrukcí. Instrukce jsou číslovány od 1 (žádný program tedy nemá instrukci číslo 0).
 - Nenastane-li žádný z případů, zaplní se všechny buňky paměti číslem 42 a program se začne vykonávat od začátku.
- ☉ \mathfrak{L} - Lev, mávaje svým ocasem, chytá tvory a zavírá je, jak jeho vůle jest.
Načte jednu hodnotu ze vstupu a uloží ji na místo v paměti určené svým rozbaleným parametrem.
- ☉ \mathfrak{M} - Klepeta štíra jsou poslední zastávkou před cestou do světa.
Rozbalí svůj parametr a vypíše jeho hodnotu na výstup.
- ☉ \mathfrak{Y} - Jeho rohy rozcestí jest. Doprava či doleva, pravá cesta vede?
Rozbalí svůj parametr a interpretuje výsledek jako adresu. Pokud je na této adrese uložena nula, následující instrukce se přeskočí. Pokud je instrukce \mathfrak{Y} s parametrem P poslední v programu (a nebylo by co přeskakovat), interpretuje se místo toho jako instrukce \nearrow s parametrem $\mathfrak{A}P$.