

SuperSopa

ALGORÍSMIA

Marc Asenjo i Ponce de León

Joan Marcè i Igual

17 de juny de 2016



**Facultat d'Informàtica
de Barcelona**

Índex

1	Introducció	3
1.1	Influències	3
2	Principals característiques	3
2.1	Paradigma	3
2.2	Compilat o interpretat?	3
2.3	Comprovació de tipus	3
2.4	Sistema de pila	4
3	Exemples	4
4	Llenguatges similars	5

1 Introducció

FORTH va ser dissenyat per *Charles Chuck Moore* [?]. Va ser creat als voltants de 1968 orientat a la indústria aeroespacial [?] i per ser integrat a aplicacions en temps real per sistemes que no tenien un sistema operatiu molt formal.

El nom fa referència a que havia de ser la quarta generació de llenguatges[?] (ve de *fourth* en anglès però la *u* no es va posar perquè el SO utilitzat només suportava 5 caràcters per als noms d'arxius).

Cap als anys 1980 es va fer força popular ja que era força fàcil de portar a petits computadors de l'època ja que era molt compacte i portable.

1.1 Influències

FORTH va ser influenciat pels següents llenguatges de programació [?]:

- Burroughs large systems
- Lisp
- APL

2 Principals característiques

2.1 Paradigma

FORTH funciona amb un paradigma **Imperatiu** i utilitza un **sistema de pila** per a les operacions. Aquest sistema de pila és diferent al d'altres llenguatges de programació que solen utilitzar-lo pel control de funcions recursives o per emmagatzemar les variables locals de la subrutina, això fa que moltes de les instruccions de FORTH siguin simplement per operar directament amb les dades de la pila.

2.2 Compilat o interpretat?

El llenguatge està pensat perquè sigui **interpretat** tot i que hi ha la opció de compilar certes comandes per a usar-les posteriorment. El sistema funciona de manera que sigui una consola interactiva però si s'han definit fitxers amb tot de paraules(subrutines) del llenguatge aquestes seran compilades per poder-les usar a la consola de FORTH.

2.3 Comprovació de tipus

Pel que fa al sistema de tipus hi ha diferents tipus però no es fa comprovació de tipus, és a dir, si el programador s'equivoca al cridar una funció, per exemple si es passa un *float* en lloc de un *int* i es crida la funció **+** se sumaran els dos valors com si fossin *int* i les dades seran interpretades erròniament. Així que segons els creadors del llenguatge comprovar els tipus és feina del programador i no de l'interpret.

Així doncs si es volen sumar dos nombres en coma flotant caldrà cridar la funció **f+**, també hi ha les funcions **c+**, **u+** per sumar caràcters i enters sense signe respectivament. D'aquesta manera es pot veure que degut a la no comprovació de tipus tampoc hi ha sobrecàrrega d'operadors.

2.4 Sistema de pila

Com ja s'ha dit abans, FORTH funciona amb un sistema de pila. Això fa que totes les operacions estiguin orientades al voltant de la pila.

En el següent exemple a la Figura 1 es pot veure que s'han inserit els valors 10 i 20. A continuació s'executa la operació * per multiplicar els dos elements que hi ha inserits a la pila (Figura 2) i el resultat es guarda a sobre de tot de la pila (els dos elements afegits anteriorment s'han eliminat). S'afegeix el 50 a la pila (Figura 3) i s'executa la operació + (Figura 4) obtenint així a sobre de tot de la pila el valor esperat (300).

```
\Exemple per realitzar (25 * 10) + 50
25 10 * 50 +
```

```
300 ok
```

3 Exemples

A continuació es mostren una sèrie d'exemples escrits en FORTH[?]:

Aquest és l'exemple més bàsic, s'insereix el caràcter "*" a la pila i s'utilitza la comanda EMIT perquè s'imprimeixi per pantalla. També s'està declarant la paraula clau "STAR"

```
: STAR ( — )      \Imprimeix * per pantalla
[CHAR] * EMIT ;    \ $ STAR
                    \ *
```

En aquest exemple s'utilitza la paraula clau declarada a l'exemple anterior. S'està declarant un bucle que va des de 0 a n-1 (n és un valor que estava inserit a la pila abans de la crida a STARS).

```
: STARS      ( n — )    \Imprimeix n estrelles
0 DO STAR LOOP      \Fer n iteracions (de 0 a n-1) i executa
STAR ;              \ $ 3 STARS
                    \ ***
```

En aquest exemple també s'utilitza la paraula clau declarada anteriorment tot i que aquí el que s'està fent és posar un bucle dins d'un bucle mitjançant la comanda DUP que duplica el valor de sobre de tot de la pila. Així doncs es duplica n per poder fer un bucle de 0 a n-1 i cridar STARS amb el paràmetre n.

```
: SQUARE      ( n — )    \Imprimir n^2 estrelles en forma de quadrat
DUP 0 DO      \ $ 3 SQUARE
DUP STARS CR  \ ***
LOOP DROP ;   \ ***
              \ ***
```

En aquest exemple s'utilitza la paraula clau STARS per escriure una línia de *. Aquí però també s'utilitza la paraula clau I que permet accedir al comptador dins d'un bucle. Per poder realitzar el bucle correctament aquí primerament s'incrementa en 1 el valor n que hi havia a la pila i es realitza el bucle de 1 a n.

```
: TRIANGLE    ( n — )    \Imprimir un triangle de n-línies
1 + 1 DO      \ $ 3 TRIANGLE
I STARS CR    \ *
LOOP ;        \ **
              \ ***
```

4 Llenguatges similars

A partir de FORTH van aparèixer altres llenguatges que utilitzaven sistemes similars. Un d'ells és el RPL que utilitzen avui dia les calculadores programables HP. També ha inspirat a *PostScript* que s'utilitza en la creació de documents. Altres han estat *Factor* o *REBOL*.