



**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE CÓMPUTO**



COMPILADORES

3CM7

---

# PRÁCTICA 04

---

**Alumno:**

Díaz Medina Jesús Kaimorts

**Profesor:**

Tecla Parra Roberto

Máquina de Pila

29 de noviembre 2018

## Introducción.

Una máquina de pila es un modelo computacional en el cual la memoria de la computadora toma la forma de una o más pilas. El término también se refiere a un computador real implementando o simulando una máquina de pila idealizada.

Adicionalmente, una máquina de pila también puede referirse a una máquina verdadera o simulada con un conjunto de instrucciones de "0 operandos". En tal máquina, la mayoría de las instrucciones implícitamente operan en valores en el tope de la pila y reemplazan esos valores por el resultado. Típicamente tales máquinas también tienen una instrucción "load" y una instrucción "store" que leen y escriben a posiciones arbitrarias de la RAM. (Como el resto de las instrucciones, las instrucciones "load" y "store" no necesitan ningún operando en una máquina de pila típica - ellas siempre toman la dirección de la RAM que se quiere leer o escribir desde el tope de la pila).

La ventaja de las máquinas de pila ("conjunto de instrucciones de 0 operandos") sobre las máquinas de acumulador ("conjunto de instrucciones de 1 operando") y las máquinas de registro ("conjunto de instrucciones de 2 operandos" o un "conjunto de instrucciones de 3 operandos") es que los programas escritos para un conjunto de instrucciones de "0 operandos" generalmente tienen una densidad de código más alta que los programas equivalentes escritos para otros conjuntos de instrucciones

## Desarrollo.

Como hemos estado manejando YACC en las anteriores prácticas, esta vez volveremos a subir de nivel.

Esta ocasión se añadirá una máquina de pila virtual a lo que se hizo en la práctica 3. Esta se encargará de evaluar a los vectores de tal manera que se irán guardando en la pila, pero en caso de necesitar una operación se realizará y el resultado también será guardado en ella.

En esta práctica tuvieron que modificarse varias cosas para poder adaptarla al hoc4, que es en donde se introdujo a la máquina virtual de pila.

Primeramente, se tuvo la aparición de las funciones como varpush y constpush que meterán valores a la pila.

```

#define STOP (Inst) 0
extern Inst prog[];

extern void assign();
extern void varpush();
extern void constpush();
extern void print();
extern void printd();
extern void constpushd();

extern void eval();
extern void add();
extern void sub();
extern void producto_cruz();
extern void producto_punto();
extern void magnitud();
extern void escalar();

```

Posteriormente en donde se encontraban estas funciones había que modificar el ingreso de los vectores, ya que funcionaban exclusivamente con números dobles, y entonces había que usar las funciones de las operaciones sobre vectores.

```

/*Seccion de Reglas Gramaticales y Acciones*/
%%
list:
| list '\n'
| list asgn '\n' { code2(pop, STOP); return 1; }
| list expr '\n' { code2(print, STOP); return 1; }
| list escalar '\n' { code2(printd, STOP) return 1; }
| list error '\n' { yyerror; }
;
asgn: VAR '=' expr { code3(varpush, (Inst)$1, assign); }
;
expr: vector { code2(constpush, (Inst)$1); }
| VAR { code3(varpush, (Inst)$1, eval); }
| asgn
| expr '+' expr { code(add); }
| expr '-' expr { code(sub); }
| escalar '*' expr { code(escalar); }
| expr '*' escalar { code(escalar); }
| expr '#' expr { code(producto_cruz); }
;
escalar: numero { code2(constpushd, (Inst)$1); }
| expr ':' expr { code(producto_punto); }
| '|' expr '|' { code(magnitud); }
;
vector: '[' NUMBER NUMBER NUMBER ']' { Vector * vector1 = creaVector(3);
vector1->vec[0] = $2;
vector1->vec[1] = $3;
vector1->vec[2] = $4;
$$ = install("", VECT , vector1);
}
;
numero: NUMBER { $$ = installd("", NUMB, $1); }
%%

```

```

void constpush() {          /* Meter una constante en la pila*/
    Datum d;

```

```

    d.val = ((Symbol *) *pc++)->u.val;    /* Apunta a la entada de
                                           la tabla de simbolos */
    push(d);
}

void constpushd() {                      /* Meter una constante en la pila*/
    Datum d;
    d.num = ((Symbol *) *pc++)->u.num;    /* Apunta a la entada de
                                           la tabla de simbolos */
    push(d);
}

void varpush() {                        /* Meter una variable a la pila */
    Datum d;                            /* Los elementos de la maquina virtual
                                           de la pila son de tipo Datum */
    d.sym = (Symbol *) (*pc++);          /* Convertimos a Symbol, lo guardamos
                                           en .sym y lo metemos en la pila */
    push(d);
}

```

```

vec1=[1 0 0]
vec2=[0 1 0]

vec1+vec2
[1.000000, 1.000000, 0.000000]

vec1-vec2
[1.000000, -1.000000, 0.000000]

5*vec1
[5.000000, 0.000000, 0.000000]

|vec1|
1.000000

|vec2|
1.000000

vec1#vec2
[0.000000, 0.000000, 1.000000]

vec1:vec2
0.000000

```

## Conclusión.

Una máquina de pila virtual nos permite realizar muchas cosas, entre ellas la calculadora científica ya que se usa para ir guardando los números y posteriormente el resultado de sus operaciones, siendo esto la principal diferencia respecto a la práctica 3 que no usa una máquina de pila virtual.