**Práctica Nro. 5 Pilas de ejecución**

**Objetivo:** Interpretar cómo se organiza la memoria de datos durante la ejecución de un programa con llamados a subrutinas.

**Ejercicio 1:** Explique claramente cual es la utilidad del registro de activación y que representan cada una de sus partes.(Basado en el modelo debajo detallado)

Modelo de registro de activación

|  |
| --- |
| Head (prog principal) |
| Pto retorno |
| EE (enlace estático) |
| ED (enlace dinámico) |
| Variables... |
| ... |
| Parámetros ... |
| …. |
| Procedimientos …. |
| …. |
| Funciones ... |
| …. |
| Valor de retorno |

RTA {

- HEAD

o Current: Dirección base del registro de activación de la unidad que se esté

ejecutando actualmente

o Free: Próxima dirección libre en la pila

- Punto de retorno: la siguiente dirección a ejecutar después de que termine la subrutina

- Enlace estático: apunta al registro de activación de la unidad que estáticamente la

contiene

- Enlace dinámico: Contiene un puntero a la dirección base del registro de activación de

la rutina llamadora

- Variables: las variables definidas dentro de la unidad.

- Procedimientos: los procedimientos definidos dentro de la unidad.

- Funciones: las funciones definidas dentro de la unidad.

- Valor de retorno: Al terminar una rutina se desaloca su RA, por lo tanto la rutina

llamante debe guardar en su RA el valor de retorno de la rutina llamada.

}

**Ejercicio 2:** Dado el siguiente programa escrito en Pascal-like, continuar la realización de las pilas de ejecución hasta finalizar las mismas.

a) Siguiendo la cadena estática b) Siguiendo la cadena dinámica

|  |  |
| --- | --- |
| Program Main  Var a: array[1..10] of integer; x,y,z:integer  Procedure A ()  var y,t: integer; begin  a(1):= a(1)+1;z:=z+1; t:=1; y:=2;  B(); a(y):=a(y)+3; y:=y+1;  If z=11 Then Begin  a(z-1):=a(z-2) + 3;  z:=z-4;  a(z-y):=a(z) – a(y) + 5; End;  end;  Function t():integer begin  y:=y+1; z:=z-6;  return(y+x); end; | Procedure B()  var d:integer; Procedure I () begin  x:=0; x:=x+6;  end; begin  x:=x+t; d:=0;  while x>d do begin  I(); x:=x-1; d:=d + 2;  end;  end;  begin  For x:=1 To 10 do a(x):=x; x:=5; y:=1; z:=10;  A();  For x:=1 To 10 do write(a(x),x);  end. |

Nota: La forma de evaluación de este lenguaje es de izquierda a derecha

Siguiendo la cadena estática Siguiendo la cadena dinámica

|  |  |
| --- | --- |
|  | \*\*\* Reg Activ Main |
| \*1 | Pto retorno |
|  | A(1)= ~~1~~ |
|  | A(2)= 2 |
|  | A(3)= 3 |
|  | A(4)= 4 |
|  | A(5)= 5 |
|  | A(6)= 6 |
|  | A(7)= 7 |
|  | A(8)= 8 |
|  | A(9)= 9 |
|  | A(10)= 10 |
|  | X= ~~1..10~~ 5 |
|  | Y= ~~1~~ – 2 |
|  | Z=~~10~~ – ~~11~~ – 5 |
|  | Procedure A |
|  | Function T |
|  | Procedure B |
|  | VR . . . . . . . . |
| \*2 | \*\*\*Reg Activ A |
|  | Pto Retorno |
|  | EE (\*1) |
|  | ED (\*1) |
|  | Y = 2 |
|  | T = 1 |
|  | VR . . . . . . . . |
|  | \*\*\* Reg Activ B |
|  | Pto Retorno |
|  | EE |
|  | ED |
|  | D = |
|  | Procedure I |
|  | VR . . ¿? . . |
|  | \*\*\* Reg Activ…(a partir de  acá lo debe continuar… |
|  | …………………… |

|  |  |
| --- | --- |
|  | \*\*\* Reg Activ Main |
| \*1 | Pto retorno |
|  | A(1)= ~~1, 2,~~ 5 |
|  | A(2)= 2 |
|  | A(3)= 3 |
|  | A(4)= 4 |
|  | A(5)= 5 |
|  | A(6)= 6 |
|  | A(7)= 7 |
|  | A(8)= 8 |
|  | A(9)= 9 |
|  | A(10)= 10 |
|  | X= ~~1..10~~ 5 |
|  | Y= ~~1~~ – 2 |
|  | Z=~~10~~ – 11 |
|  | Procedure A |
|  | Function T |
|  | Procedure B |
|  | VR . . . . . . . . |
| \*2 | \*\*\*Reg Activ A |
|  | Pto Retorno |
|  | EE (\*1) |
|  | ED (\*1) |
|  | Y = 2 |
|  | T = 1 |
|  | VR . . . . . . . . |
| \*3 | \*\*\* Reg Activ B |
|  | Pto Retorno |
|  | EE (\*1) |
|  | ED (\*2) |
|  | D = |
|  | Procedure I |
|  | VR . . ¿? . . |
| \*4 | \*\*\* Reg Activ…(a partir de  acá lo debe continuar… |
|  | …………………… |

**Ejercicio 3:** Sea el siguiente programa escrito en Pascal-like. Realice la pila de ejecución

1. Siguiendo la cadena estática
2. Siguiendo la cadena dinámica

|  |  |
| --- | --- |
| PROGRAM P1;  var  a:integer; b:char;  c: array[1..10] of integer  Procedure PP1; var  a:char; p:integer;  Function x: integer; var  z:integer;  begin  a:="j"; z=-1;  return z;  end;  Begin  p:=x; write(a); p:=x+3; c[p]=8; p:=x+2;  c[p]=x;  end; | Procedure x; var  b:char; Procedure PP2;  Begin  write("para qué estoy aquí?");  end;  Begin  a:=1; c[a]:=4;  b:="a";  write(concat(c[1],b)); /\*concat convierte a string los parámetros, concatena y retorna un string;\*/  PP1();  b:="b";  write(concat(c[5],b)); /\*concat convierte a string los parámetros, concatena y retorna un string;\*/  End; BEGIN a:=3; b:="c";  for a:=3 to 10 do  begin  c[a]:=2\*a;  end;  x; write(b);  write(a);  for a:=1 to 10 do  write(c[a]-3);  END. |

Nota: La forma de evaluación de este lenguaje es de izquierda a derecha

**Ejercicio 4:** Sea el siguiente programa escrito en Pascal-like. Realice la pila de ejecución

1. Siguiendo la cadena estática
2. Siguiendo la cadena dinámica

|  |  |
| --- | --- |
| Procedure Main;  var x, y: integer;  vec: array[1..7] of integer; Function B:integer;  var y:integer; begin  y:=4; x:= y - 2;  return (x); end;  Procedure D;  var i, x: integer;  vec: array[1..7] of integer; Procedure A;  var y:integer; begin  y:=x + 5; vec(i + 2):= vec(i + 2) + y;  x:= x +B; C;  end; Function B:integer;  begin  vec(i):= y + 2; i:=i+2;  vec(i):= vec(1) \* i;  return ( vec(i)-vec(1) ); end;  begin  for x:= 1 to 7 do vec(x):= 1;  x:=1; i:= 2;  if y = 7 then A; else C;  for x:= 1 to 7 do write(vec(x)); end; | Procedure C;  var i, y: integer; begin  i:= 1; y:= 6; x:= x + B;  vec(2):= vec(2) \* x; while (i < y) do begin  vec(i):= vec(i) + B - 1; i:= i + 3;  end;  y:= y - 4;  end;  begin  for x:= 1 to 7 do vec(x):= x; x:= 3; y:= B+5; D;  if (x = 2) then begin vec(x):= vec(x) + 2;  vec(x + 3):= vec(x) \* 3; end;  for x:= 1 to 7 do write(vec(x)); end. |

Nota: La forma de evaluación de este lenguaje es de izquierda a derecha

**Ejercicio 5:** Sea el siguiente programa escrito en Pascal-like. Realice la pila de ejecución

1. Siguiendo la cadena estática
2. Siguiendo la cadena dinámica
3. La sentencia x:= c + 5 +x, podría reemplazarse por x:= x + c + 5? Justifique la respuesta

|  |  |
| --- | --- |
| Program Main;  Var x, y, z:integer;  a, b: array[1..6] of integer; Procedure B;  var y,x: integer; Procedure C;  var c:integer; begin  y:= y + 2; c:=2;  a(x):=a(x)\*y; if (y >7) then  b(y-6)=b(4)\*2+b(y  -6); D;  end; begin  x:=2; y:= x + 3;  C; x:= x + 1; write (x,y); End;  Procedure D;  begin  x:= c + 5 + x; y:= y + 2;  end; | Function C: integer; begin  b(x):= b(x) + 1; x:= x + 1;  a(y):=a(y)+b(x)+3; a(x+2)=a(x) + 2;  return b(x); end  begin  x:= 1; Y:= 2;  for z:=1 to 6 do begin a(z):= z;  b(z):= z + 2; end;  B;  for z:= to 6 do write (a(z), b(z)); end. |

Nota:La forma de evaluación de este lenguaje es de izquierda a derecha