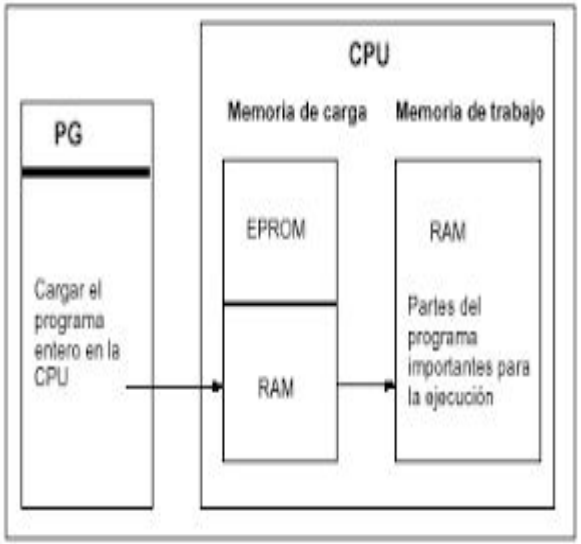
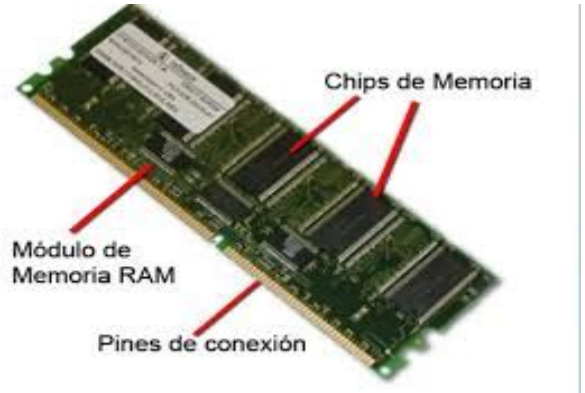
Memoria

Memoria

Todo archivo escrito en código máquina se lo suele llamar Programa. Pero un programa se convierte en tal cuando es copiado a la memoria RAM y se le han suministrado todos los recursos que requiera para su funcionamiento.

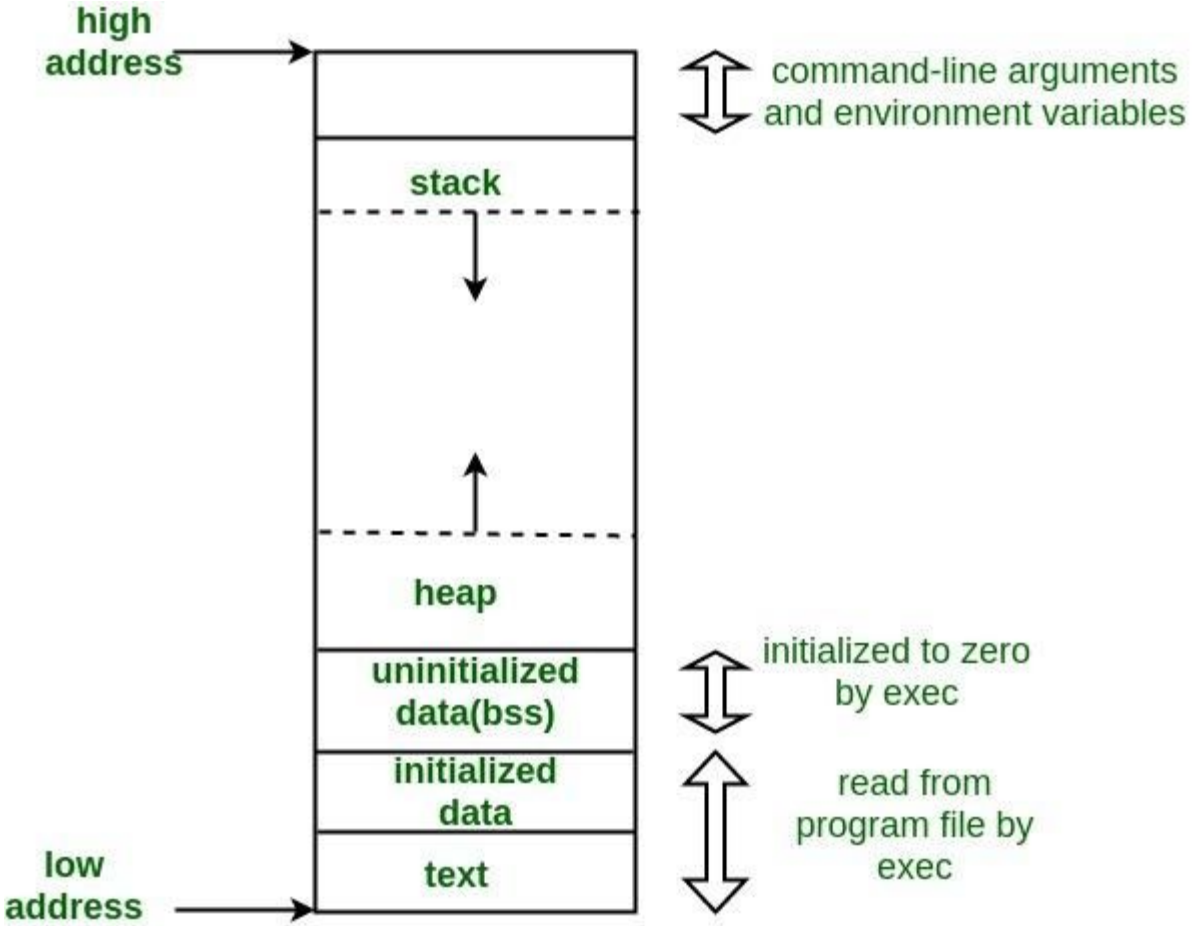
“Un Programa solo existe cuando se encuentra en ejecución en la memoria RAM”





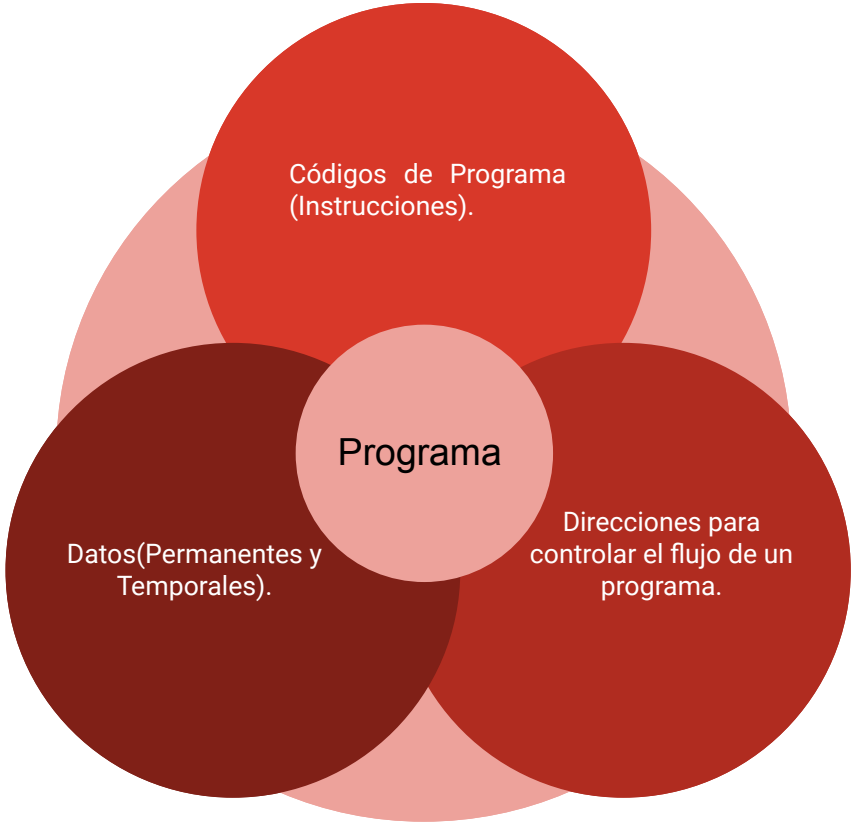
Memoria

Desde la sección baja de memoria se copia el programa, se definen las variables estáticas . En la parte superior de la memoria se copian los llamados a funciones y sus variables. El sector libre denominado Memoria de Montículo es la sección que se emplea para variables dinámicas.



Ejecución de un Programa

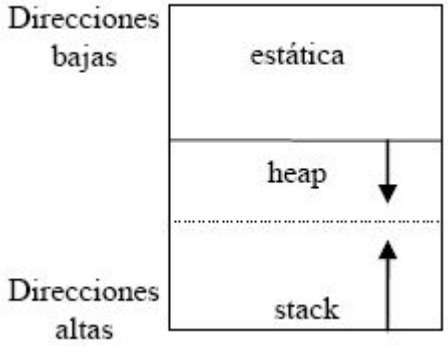
La ejecución de un programa requiere que diversos elementos se almacenen en memoria:



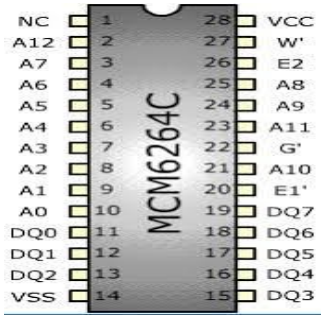
Memoria Estática y Memoria Dinámica

La asignación de memoria para algunos elementos fijos del programa que es controlado por el compilador se llama Memoria Estática.

A la asignación y posible recuperación de memoria durante la ejecución de un programa y bajo su control, se la llama memoria dinámica.



Organización de sección de Memoria.



Circuito integrado de Memoria RAM

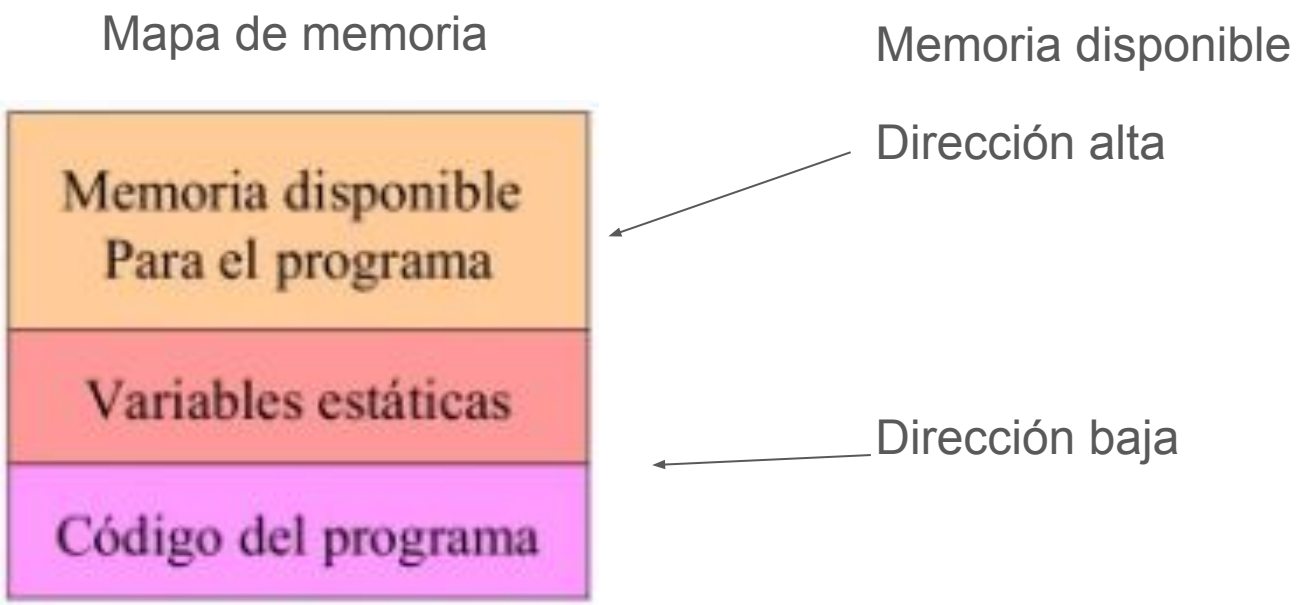
Memoria Estática

Elementos que residen en memoria estática:

- Código del programa
- Las variables definidas en la sección principal del programa, las cuales pueden solo cambiar su contenido no su tamaño.
- Todas aquellas variables declaradas como estáticas en otras clases o módulos.

Estos elementos se almacenan en direcciones fijas que son relocalizadas dependiendo de la dirección en donde el cargador las coloque para su ejecución.

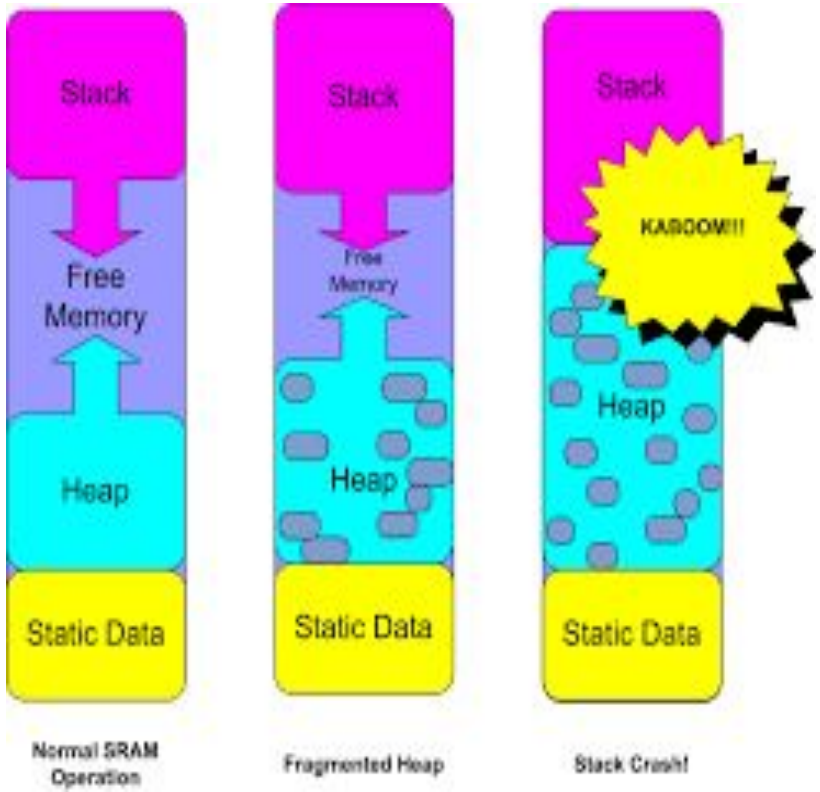
Método común de asignación de memoria



El stack de ejecución

- Cada subprograma (procedimiento, función, método, etc.) requiere una representación de sí en tiempo de ejecución.

- Estas representaciones se almacenan en el stack de ejecución con el fin de controlar el flujo de ejecución del programa.



Memoria Dinámica

Define el tamaño del espacio de memoria necesario para un programa en tiempo de ejecución.

El tamaño de los elementos puede cambiar durante la ejecución del programa.

Almacena todos los elementos definidos con la palabra new en un programa.

Definimos un Vector Dinámico

A continuación se emplea un array entero unidimensional y se asigna al puntero . Los elementos del array se generan con la función new y se liberan con la función delete.

Test:
https://replit.com/join/dzyntlukxs-maximo-arielari

```
int *p;

int main() {
  int u=0;
  cout << "Sistema de Gestión de Memoria\n";
  do{
  cout << "Cuantos elementos desea ingresar :\n";
  cin>>u;
    }while(u==0);
  p= new int[u];
  cout<<"\n Tamaño de puntero :"<<sizeof(*p);

  delete[]p;

}
```

Ejemplo: (Ingreso- Listado)

Vector definido mediante técnica de asignación dinámica según los conceptos vistos.

```
p= new int[u];
```

Test:

https://replit.com/join/sjkhxgnklc-maximo-arielari