

Tutorial de Tomos

<https://github.com/jmansilla/tomos>

Tomos es un visualizador de ejecuciones de pequeños programas escritos en el lenguaje del teórico/práctico de la materia Algoritmos II.

Es especialmente útil para entender y debuggear algoritmos que manipulan memoria dinámica usando punteros.

En la versión actual, Tomos sólo soporta declaraciones de tipos, declaraciones de variables y secuencias de sentencias. No se pueden definir procedimientos ni funciones. Tampoco se pueden especificar ni implementar TADs.

Instalación

Se recomienda instalar en un entorno virtual de python usando una terminal de linux. Primero se debe crear el entorno virtual usando [venv](#). Por ejemplo:

```
$ python3.12 -m venv myenv
```

Ahora tenemos un entorno virtual en la carpeta myenv.

Para instalar Tomos ejecutar:

```
$ ./myenv/bin/pip install tomos
```

Si se instaló correctamente, tendremos disponible el comando `tomos` dentro del entorno virtual. Probar ejecutando:

```
$ ./myenv/bin/tomos --version  
Tomos version 0.1.5
```

Coloreador de sintaxis en VSCODE: descargar la última versión disponible [acá](#) (archivos vsix). Para instalar un archivo VSIX en VS Code, abrir la vista de Extensiones (Ctrl+Shift+X), selecciona "Instalar desde VSIX" y elegir el archivo VSIX que deseas instalar. Luego, el editor puede solicitar que se reinicie para aplicar la nueva extensión.

Ejemplos: Listas enlazadas

https://github.com/jmansilla/tomos/tree/main/demo/linked_list

Setup: Construyendo una lista enlazada

Descargar el archivo:


https://github.com/jmansilla/tomos/blob/main/demo/linked_list/setup.ayed

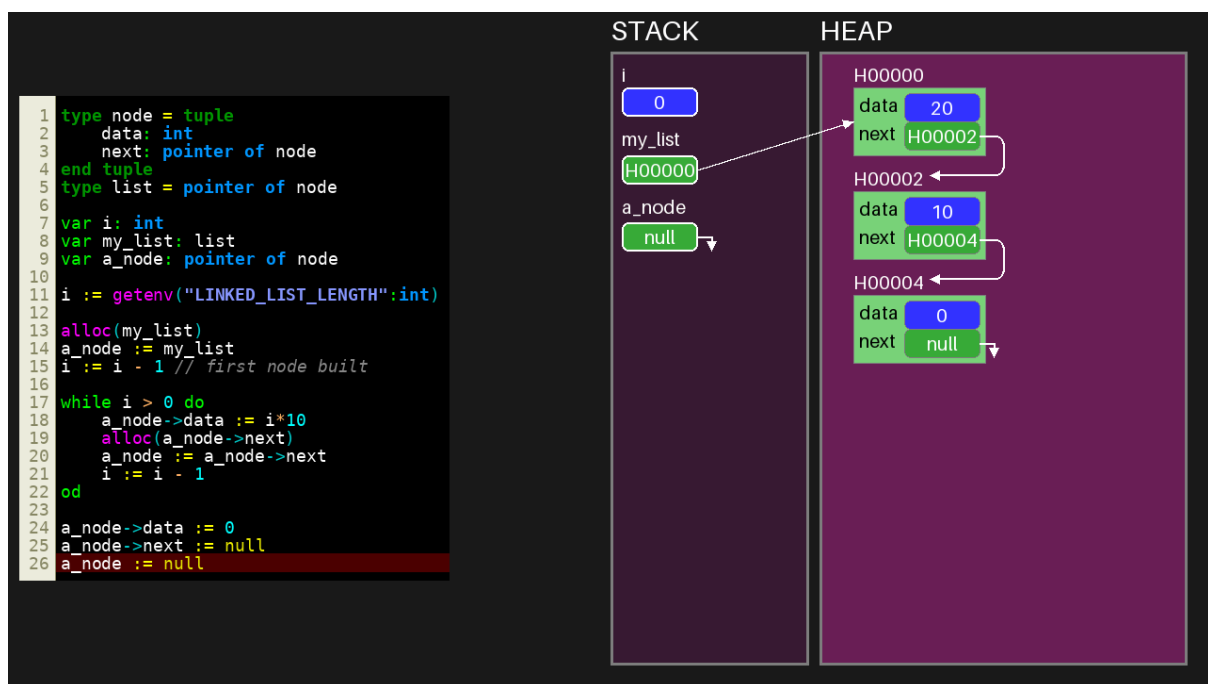
Este código construye una lista enlazada con una cantidad de elementos indicada por la variable de entorno LINKED_LIST_LENGTH. Leerlo y asegurarse de **entenderlo por completo**. ¿Qué valores tienen los elementos de la lista?

Ejecutar el código con siguiente comando de Tomos:

```
$ export LINKED_LIST_LENGTH=3
$ ./myenv/bin/tomos setup.ayed --save-state=setup.st --movie=setup.mp4
```

Este comando ejecuta el código para crear una lista de tres elementos. El estado final se guarda en el archivo setup.st, para ser usado después como estado inicial de los ejemplos siguientes.

La visualización se guarda en el archivo de video  setup.mp4 . **Mirar con detenimiento el video y entender perfectamente qué sucede en cada paso.** El estado final se ve así:



Append: Agregando un elemento al final de la lista

Descargar el archivo:


https://github.com/jmansilla/tomos/blob/main/demo/linked_list/append.ayed

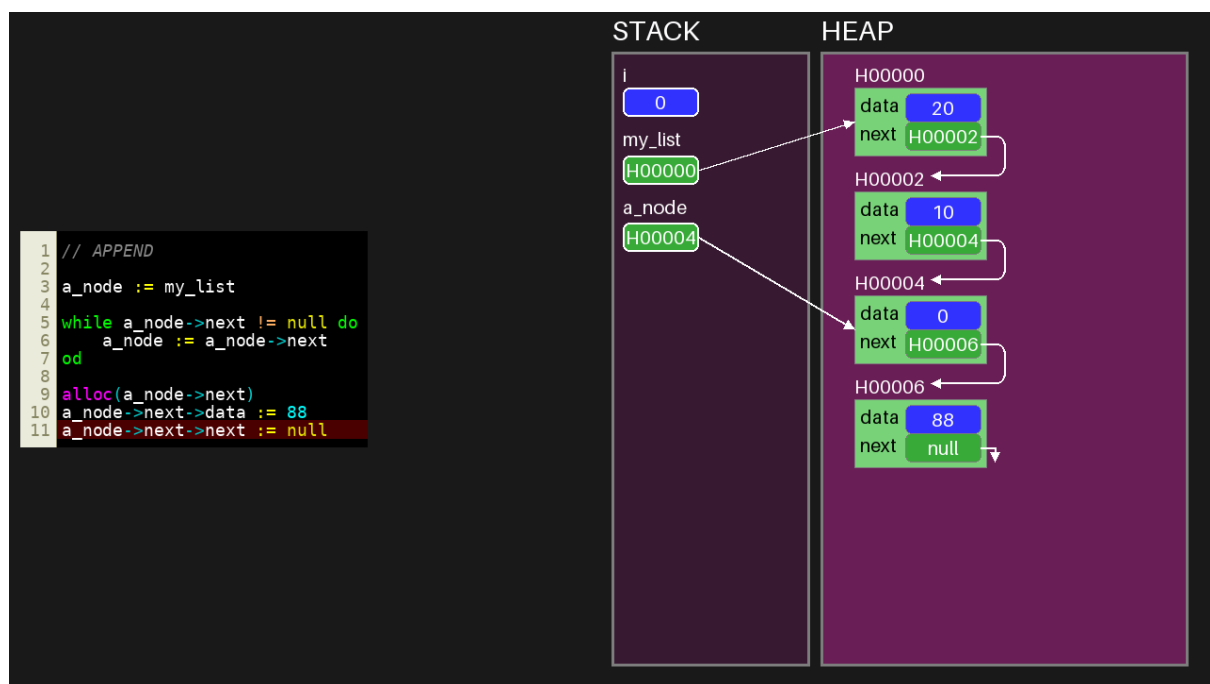
Este código agrega un elemento al final de una lista enlazada que debe ser cargada a partir de un estado inicial.

Ejecutar el código con siguiente comando de Tomos:

```
$ ./myenv/bin/tomos tomos/demo/linked_list/append.ayed  
--load-state=setup.st --movie=append.mp4
```

Al igual que en el ejemplo anterior, se genera la visualización en un archivo de video

 append.mp4 . Nuevamente, **mirar con detenimiento el video y entender perfectamente qué sucede en cada paso**. El estado final se ve así:



Más ejemplos

Repetir el mismo proceso con los siguientes archivos:

https://github.com/jmansilla/tomos/blob/main/demo/linked_list/tail.ayed

https://github.com/jmansilla/tomos/blob/main/demo/linked_list/insert_at.ayed

https://github.com/jmansilla/tomos/blob/main/demo/linked_list/remove_first_data.ayed

https://github.com/jmansilla/tomos/blob/main/demo/linked_list/remove_not_found.ayed