

Minishell Memory Freeing Checklist

This checklist ensures all dynamically allocated memory used during the lifetime of Minishell is properly freed before the program exits. This includes tokens, AST, redirections, environment, and user input.

- [] Implement `free_minishell(t_minishell *mini)`
- [] Free token list (`t_list *token_list`)
 - [] Free each `t_token` value
 - [] Free list nodes
- [] Free AST recursively (`t_command_tree *`)
 - [] Traverse left and right nodes
 - [] Free `t_exec` inside `N_EXEC` nodes
 - [] Free infiles and outfiles lists
- [] Free environment list (`t_list *envp`)
 - [] Free each `t_env` key/value
- [] Free `envp_arr` (`char **`)
- [] Free input string (`mini->input`)
- [] Free current working directory (`mini->cwd`)
- [] Nullify all freed pointers to avoid dangling access
- [] Handle cleanup if execution was interrupted (e.g., Ctrl-C)
- [] Free any heredoc temporary files if created
- [] Ensure no memory leaks (test with `valgrind`)

Use this checklist to guarantee clean program termination and leak-free memory.