```
// Initialize Minnie's file system
function initializeFileSystem() {
  console.log("Initializing Minnie's file system...");

  // Create necessary directories
  createDirectoryIfNotExists("/knowledge_base/");
  createDirectoryIfNotExists("/knowledge_base/compressed/");
  createDirectoryIfNotExists("/cache_files/");
  createDirectoryIfNotExists("/cache_files/compressed/");
  createDirectoryIfNotExists("/ghost_files/");

  // Check for existing files and their compression status
  updateFileStatus();

  // Decompress essential files for quick startup
  const essentialFiles = ["general_vocabulary", "commands"];
  for (const fileId of essentialFiles) {
    decompressFile(fileId);
    decompressFile(`${fileId}_cache`);
  }

  console.log("File system initialized successfully");
}

// Update the status of all files (compressed/uncompressed)
function updateFileStatus() {
  // Check knowledge base files
  for (const [fileId, fileDetails] of
Object.entries(this.knowledge_base.files)) {
    fileDetails.uncompressed_available =
fileExists(fileDetails.path);
    fileDetails.is_compressed =
fileExists(fileDetails.compressed_path);
  }

  // Check cache files
  for (const [fileId, fileDetails] of
Object.entries(this.cache_files.files)) {
    fileDetails.uncompressed_available =
fileExists(fileDetails.path);
    fileDetails.is_compressed =
fileExists(fileDetails.compressed_path);
  }
}

// Start up Minnie
function startup() {
  console.log("Starting Minnie...");

  // Initialize file system
  initializeFileSystem();

  // Set up any scheduled tasks
  if (this.file_management.auto_compress_low_priority) {
    setInterval(compressAllInactive, 300000); // Every 5 minutes
  }

  console.log("Minnie is ready");
}
```