

- 操作指令集合
- 1. 特殊命令
 - 2. 系统命令
 - 3. 文件系统命令
 - 测试示例
 - 完整文件系统测试流程
 - 4. 进程管理命令
 - 可运行的程序
 - 测试示例
 - 5. 开发工具命令
 - 5.1 汇编器命令
 - 5.2 实验演示命令
 - 测试示例
 - 6. AI模式测试
 - 6.1 进入AI模式
 - 6.2 AI自然语言命令测试
 - 6.3 AI问答测试
 - 6.4 混合使用测试
 - 7. 完整测试流程
 - 批量命令测试

操作指令集合

1. 特殊命令

以 / 开头的命令，在任何模式下都可用

命令	说明	示例
/help	显示命令帮助	/help
/mode	切换AI模式/普通模式	/mode
/cmd <命令>	在AI模式下直接执行系统命令	/cmd ls
/clear	清空AI对话历史	/clear
/exit	退出系统	/exit

2. 系统命令

命令	说明	参数
help	显示帮助信息	无
sysinfo	显示系统信息	无
cpuinfo	显示CPU状态和寄存器	无
meminfo	显示内存使用情况	无
fsinfo	显示文件系统信息	无
ai	获取AI系统分析报告	无
clear	清屏	无
exit / quit	退出系统	无

3. 文件系统命令

命令	说明	参数	示例
ls	列出目录内容	[路径]	ls 或 ls /home
cd	切换目录	<路径>	cd /home
pwd	显示当前目录	无	pwd
mkdir	创建目录	<路径>	mkdir /test
touch	创建文件	<路径>	touch /test.txt
cat	读取文件内容	<路径>	cat /test.txt
echo	写入文件	<路径> <内容>	echo /test.txt Hello
rm	删除文件/目录	<路径>	rm /test.txt

测试示例

```
pwd
ls
```

```
mkdir /home
mkdir /home/user
cd /home/user
pwd
touch readme.txt
echo readme.txt Hello LZY-OS!
cat readme.txt
ls
cd /
ls /home/user
rm /home/user/readme.txt
```

完整文件系统测试流程

```
# 1. 查看当前位置
pwd

# 2. 创建目录结构
mkdir /projects
mkdir /projects/demo
cd /projects/demo

# 3. 创建和编辑文件
touch hello.txt
echo hello.txt Welcome to LZY-OS!
cat hello.txt

# 4. 查看目录内容
ls
ls /projects

# 5. 返回根目录
cd /
pwd

# 6. 查看文件系统信息
fsinfo
```

4. 进程管理命令

命令	说明	参数	示例
run	运行程序	<程序名>	run fibonacci
ps	显示进程信息	无	ps

可运行的程序

程序名	说明
fibonacci	斐波那契数列程序
sum	求和程序
hello	Hello World程序
multiply	乘法程序

测试示例

```
run fibonacci
ps
run sum
ps
run hello
un multiply
cpuinfo
```

5. 开发工具命令

5.1 汇编器命令

命令	说明	参数
asm	汇编程序	<程序名>

可汇编的程序： fibonacci, sum, hello, multiply

```
asm fibonacci
asm sum
```

5.2 实验演示命令

命令	说明
exp producer-consumer	生产者-消费者问题演示
exp memory-allocation	内存分配算法演示
exp process-scheduling	进程调度算法对比
exp filesystem	文件系统操作演示

测试示例

```
exp producer-consumer
exp memory-allocation
exp process-scheduling
exp filesystem
```

6. AI模式测试

6.1 进入AI模式

```
/mode
[系统] 已切换到：AI对话模式
```

6.2 AI自然语言命令测试

自然语言输入	预期执行命令
"帮我创建一个test目录"	mkdir /test
"查看CPU状态"	cpuinfo
"显示内存信息"	meminfo
"运行斐波那契程序"	run fibonacci
"列出当前目录"	ls

自然语言输入	预期执行命令
"演示生产者消费者问题"	<code>exp producer-consumer</code>
"查看系统信息"	<code>sysinfo</code>
"退出系统"	<code>exit</code>

6.3 AI问答测试

什么是进程调度？
解释一下内存碎片
什么是死锁？
FCFS和RR调度算法有什么区别？
什么是生产者消费者问题？

6.4 混合使用测试

帮我创建一个projects目录
`/cmd cd /projects`
创建一个readme.txt文件
往readme里写入Hello World
显示文件内容
`/mode`
`ls /projects`

7. 完整测试流程

```
# ===== 启动系统 =====  
python main.py  
  
# ===== 系统信息测试 =====  
sysinfo  
cpuinfo  
meminfo  
fsinfo  
  
# ===== 文件系统测试 =====  
pwd  
mkdir /home
```

```
mkdir /home/user
cd /home/user
touch test.txt
echo test.txt Hello GALAXY-OS
cat test.txt
ls
cd /
ls /home/user

# ===== 进程管理测试 =====
run fibonacci
run sum
ps

# ===== 汇编器测试 =====
asm fibonacci

# ===== 实验演示测试 =====
exp producer-consumer
exp memory-allocation

# ===== AI模式测试 =====
/mode
帮我创建一个demo目录
查看CPU状态
什么是进程？
/mode
ls /demo

# ===== AI分析测试 =====
ai

# ===== 退出 =====
exit
```

批量命令测试

```
sysinfo
cpuinfo
meminfo
fsinfo
pwd
mkdir /batch_test
cd /batch_test
touch file1.txt
touch file2.txt
echo file1.txt Content1
echo file2.txt Content2
cat file1.txt
ls
run fibonacci
ps
```

```
exp producer-consumer
ai
cd /
rm /batch_test/file1.txt
rm /batch_test/file2.txt
exit
```