

ANDRÉ LUCHESI
CARLOS CUNHA
CESAR HENRIQUE KÁLLAS

**SISTEMAS OPERACIONAIS II
ESPECIFICAÇÃO CAMADA INDEPENDENTE**

**Pontifícia Universidade Católica de Campinas
Faculdade de Engenharia de Computação
Turma III – Grupo 9
Outubro de 2004**

Exemplo Parcial de um Documento de Plano de Testes e de Integração

1. Abertura de um arquivo inexistente

Existem várias condições a serem testadas durante a ativação do primeiro módulo da camada independente de dispositivos (cada uma delas é detalhada abaixo, através de seu caso de teste) e que podem provocar diferentes códigos de retorno.

Retornos possíveis da função <code>ffopen</code>	
Código retorno	Descrição
inteiro positivo	Sucesso, retorna um número inteiro positivo que será usado como índice de manipulação do arquivo.
-1	Impossível abrir arquivo, arquivo não existe
-2	Erro ao iniciar estruturas de manipulação do arquivo
-3	Não foi possível comunicar com o driver
-4	Não foi possível acessar o BFD
-5	Lista de descritores cheia

Teste para abrir um arquivo inexistente

Se a camada de nível do usuário fazer uma chamada abrir um arquivo inexistente, a CID deverá retornar um código descrito na tabela acima, que descreve que o arquivo não existe (-1).

1. Condição a ser testada: caminho de arquivo inexistente, recebendo assim como retorno -1.

2. Como causar a condição:

A partir da camada ao nível do usuário (que não existe e, portanto, corresponderá a uma rotina externa para os testes), chamar a rotina relativa ao `open`: `ffopen(char *filename, int mode)` . com os seguintes parâmetros: caminho do arquivo, que será um ponteiro para uma cadeia de string de tamanho máximo de 50 caracteres.

Para isso, será criado um programa que tenha a chamada compilada junto à rotina `open`. Essa rotina é uma chamada de função, com valor de retorno inteiro positivo ou negativo.

```
char *arquivo, caminho[50];
copia_string(caminho, "naovouacharessearquivo.txt");
*arquivo=caminho;
se retorno fopen(char *arquivo, 0) = -1)
    escreva("arquivo inexistente")
```

A função `TratarOpen` e consulta a lista de arquivos abertos...

O arquivo não foi aberto, vai consultar o BFD para tentar achar o bloco inicial do arquivo percorrendo todo o caminho passado.

3. Resultado Esperado: O primeiro bloco não será encontrado e a CID irá retornar -1 à camada ao nível do usuário.

4. Como observar o resultado esperado: segue pseudo-código

Para observar o resultado:

...declarar como caminho para abertura do arquivo, um caminho inválido e/ou de um arquivo inexistente.

...o valor sera retornado para a camada do usuario, podendo intao ser impresso.

2. Abertura de um arquivo sem usar o driver

Existem várias condições a serem testadas durante a ativação do primeiro módulo da camada independente de dispositivos (cada uma delas é detalhada abaixo, através de seu caso de teste) e que podem provocar diferentes códigos de retorno.

Retornos possíveis da função <i>ffopen</i>	
Código retorno	Descrição
inteiro positivo	Sucesso, retorna um número inteiro positivo que será usado como índice de manipulação do arquivo.
-1	Impossível abrir arquivo, arquivo não existe
-2	Erro ao iniciar estruturas de manipulação do arquivo
-3	Não foi possível comunicar com o driver
-4	Não foi possível acessar o BFD
-5	Lista de descritores cheia

Teste para abrir um arquivo sem usar o driver

Se a camada de nível do usuário fazer uma chamada abrir um arquivo a CID deverá retornar um código descrito na tabela acima, (-3).

1. Condição a ser testada: erro ao comunicar com o driver, recebendo assim como retorno -3.

2. Como causar a condição:

A partir da camada ao nível do usuário (que não existe e, portanto, corresponderá a uma rotina externa para os testes), chamar a rotina relativa ao open: *ffopen(char *filename, int mode)* . com os seguintes parâmetros: caminho do arquivo, que será um ponteiro para uma cadeia de string de tamanho máximo de 50 caracteres.

Para isso, será criado um programa que tenha a chamada compilada junto à rotina open. Essa rotina é uma chamada de função, com valor de retorno inteiro positivo ou negativo.

```
char *arquivo, caminho[50];
copia_string(caminho,"c://novoarquivo.txt");
```

```
*arquivo=caminho;
se retorno fopen(char *arquivo, 1) = -4)
    escreva(“disco cheio”)
```

A função `fopen` irá fazer uma requisição ao driver e o mesmo não irá responder.

3. Resultado Esperado: A camada independente irá retornar o valor -3 à camada ao nível do usuário.

4. Como observar o resultado esperado: segue pseudo-código

Para observar o resultado:

...declarar como caminho para abertura do arquivo

...o valor será retornado para a camada do usuário, podendo então ser impresso.

3. Abertura de um arquivo com lista de descritores cheia

Existem várias condições a serem testadas durante a ativação do primeiro módulo da camada independente de dispositivos (cada uma delas é detalhada abaixo, através de seu caso de teste) e que podem provocar diferentes códigos de retorno.

Retornos possíveis da função <code>fopen</code>	
Código retorno	Descrição
inteiro positivo	Sucesso, retorna um número inteiro positivo que será usado como índice de manipulação do arquivo.
-1	Impossível abrir arquivo, arquivo não existe
-2	Erro ao iniciar estruturas de manipulação do arquivo
-3	Não foi possível comunicar com o driver
-4	Não foi possível acessar o BFD
-5	Lista de descritores cheia

Teste para abrir um arquivo com a lista de descritores cheia

Se a camada de nível do usuário fizer uma chamada para abertura de um arquivo, a CID deverá retornar um código descrito na tabela acima, -5.

1. Condição a ser testada: lista de descritores cheia, recebendo assim como retorno -5.

2. Como causar a condição:

A partir da camada ao nível do usuário (que não existe e, portanto, corresponderá a uma rotina externa para os testes), chamar a rotina relativa ao `open`: `fopen(char *filename, int mode)` . com os seguintes parâmetros: caminho do arquivo, que será um ponteiro para uma cadeia de string de tamanho máximo de 50 caracteres.

Para isso, será criado um programa que tenha a chamada compilada junto à rotina open. Essa rotina é uma chamada de função, com valor de retorno inteiro positivo ou negativo.

```
char *arquivo, caminho[50];
copia_string(caminho, "dispos://novoarquivo.txt");
*arquivo=caminho;
se retorno fopen(char *arquivo, 1) = -6)
    escreva("disco cheio")
```

A função fopen irá abrir várias vezes qualquer arquivo (minimo 10 vezes) e intão acontecerá.

3. Resultado Esperado: A camada independente irá retornar o valor -6 à camada ao nível do usuário por não achar o dispositivo.

4. Como observar o resultado esperado: segue pseudo-código

Para observar o resultado:

...declarar como caminho para abertura do arquivo

...o valor sera retornado para a camada do usuario, podendo intao ser impresso.

4. Impossível ler arquivo: sem acesso ao bfd

Existem várias condições a serem testadas durante a ativação do primeiro módulo da camada independente de dispositivos (cada uma delas é detalhada abaixo, através de seu caso de teste) e que podem provocar diferentes códigos de retorno.

Retornos possíveis da função fopen	
Código retorno	Descrição
inteiro positivo	Sucesso, retorna um número inteiro positivo que será usado como indice de manipulação do arquivo.
-1	Impossível abrir arquivo, arquivo não existe
-2	Erro ao iniciar estruturas de manipulação do arquivo
-3	Não foi possível comunicar com o driver
-4	Não foi possível acessar o BFD
-5	Lista de descritores cheia

Teste para arquivo impossível de ser lido: sem acesso ao bfd

Se a camada de nível do usuário fazer uma chamada abrir um arquivo, a CID deverá retornar um código descrito na tabela acima, -4.

1. Condição a ser testada: sem acesso ao bfd, recebendo assim como retorno -4.

2. Como causar a condição:

A partir da camada ao nível do usuário (que não existe e, portanto, corresponderá a uma rotina externa para os testes), chamar a rotina relativa ao open e apagar o diretório que contém os dados da bfd (“/bfd”).

3. Resultado Esperado: A camada independente irá retornar o valor -4 à camada ao nível do usuário.

4. Como observar o resultado esperado: segue pseudo-código

Para observar o resultado:
fazer requisição ao driver da bfd

5. Impossível ler arquivo: posição inválida do ponteiro

Existem várias condições a serem testadas durante a ativação do primeiro módulo da camada independente de dispositivos (cada uma delas é detalhada abaixo, através de seu caso de teste) e que podem provocar diferentes códigos de retorno.

Retornos possíveis da função fread	
Código retorno	Descrição
inteiro positivo	Sucesso, retorna um número inteiro positivo que é a quantidade de bytes lidos.
0	Se o ponteiro já estiver no final do arquivo, esse é um caso típico de arquivo vazio, o retorno nesse caso é visto como sucessivo.
-1	Impossível ler arquivo, corrompido.
-2	Posição inválida do ponteiro (posição > tamanho do arquivo).
-3	Dispositivo inválido.
-4	Arquivo não encontrado.

Teste para arquivo impossível de ser lido: posição inválida do ponteiro

Se a camada de nível do usuário fizer uma chamada abrir um arquivo, a CID deverá retornar um código descrito na tabela acima, que descreve uma posição inválida do ponteiro (-2).

1. Condição a ser testada: posição inválida do ponteiro, recebendo assim como retorno -2.

2. Como causar a condição:

A partir da camada ao nível do usuário (que não existe e, portanto, corresponderá a uma rotina externa para os testes), chamar a rotina relativa ao open: fread(int indice, char *buffer, int tamanho) com os seguintes parâmetros: indice de descritor, local para armazenar dados lidos, e a quantidade de dados lidos.

Para isso, será criado um programa que tenha a chamada compilada junto à rotina fread. Essa rotina é uma chamada de função, com valor de retorno inteiro positivo ou negativo.

```
Char *buffer
int tamanho 2048
```

```
se retorno fread(1232, buffer, tamanho) = -2)
    escreva("posicao invalida do ponteiro")
```

Considerando que o arquivo já tenha sido aberto antes...

A função Tratar READ/WRITE vai verificar no descritor o ponteiro para a posição atual do arquivo e o tamanho do arquivo e retornar o erro caso o primeiro ponteiro seja maior que o tamanho do arquivo.

3. Resultado Esperado: A camada independente irá retornar o valor -2 à camada ao nível do usuário.

4. Como observar o resultado esperado: segue pseudo-código

Para observar o resultado:

- ...criar arquivo vazio
- ...abrir o arquivo
- ...ler do arquivo uma posicao invalida
- ...o valor sera retornado para a camada do usuario, podendo intao ser impresso.

6. Impossível ler arquivo: estouro de deadline

Existem várias condições a serem testadas durante a ativação do primeiro módulo da camada independente de dispositivos (cada uma delas é detalhada abaixo, através de seu caso de teste) e que podem provocar diferentes códigos de retorno.

Retornos possíveis da função fread	
Código retorno	Descrição
inteiro positivo	Sucesso, retorna um número inteiro positivo que é a quantidade de bytes lidos.
0	Se o ponteiro já estiver no final do arquivo, esse é um caso típico de arquivo vazio, o retorno nesse caso é visto como sucessivo.
-1	Impossível ler arquivo
-2	Posição inválida do ponteiro (posição > tamanho do arquivo).
-3	Dispositivo inválido.
-4	Arquivo não encontrado.

Teste para arquivo impossível de ser lido: posição inválida do ponteiro

Se a camada de nível do usuário fazer uma chamada para ler um arquivo, a CID deverá retornar um código descrito na tabela acima, que descreve ser impossível ler o arquivo (-1).

1. Condição a ser testada: impossível ler arquivo, recebendo assim como retorno -1.

2. Como causar a condição:

A partir da camada ao nível do usuário (que não existe e, portanto, corresponderá a uma rotina externa para os testes), chamar a rotina relativa ao open: fread(int indice, char *buffer, int tamanho) com os seguintes parâmetros: indice de descritor, local para armazenar dados lidos, e a quantidade de dados lidos.

Para isso, será criado um programa que tenha a chamada compilada junto à rotina open. Essa rotina é uma chamada de função, com valor de retorno inteiro positivo ou negativo.

```
Char *buffer  
int tamanho 2048
```

```
se retorno fread(1232, buffer, tamanho) = -2)  
    escreva(“posicao invalida do ponteiro”)
```

Considerando que o arquivo já tenha sido aberto antes...

A função Tratar READ/WRITE vai verificar no descritor o ponteiro para a posição atual do arquivo e o tamanho do arquivo e retornar o erro caso o primeiro ponteiro seja maior que o tamanho do arquivo.

3. Resultado Esperado: A camada independente irá retornar o valor -2 à camada ao nível do usuário.

4. Como observar o resultado esperado: segue pseudo-código

Para observar o resultado:

- ...criar arquivo vazio
- ...abrir o arquivo
- ...ler do arquivo uma posicao invalida
- ...o valor sera retornado para a camada do usuario, podendo intao ser impresso.