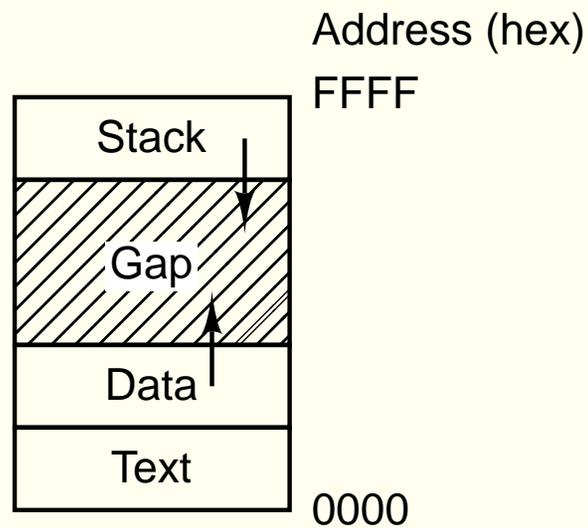


**MO806/MC914**  
**Tópicos em Sistemas Operacionais**  
2s2006

**Processos e Threads 1**

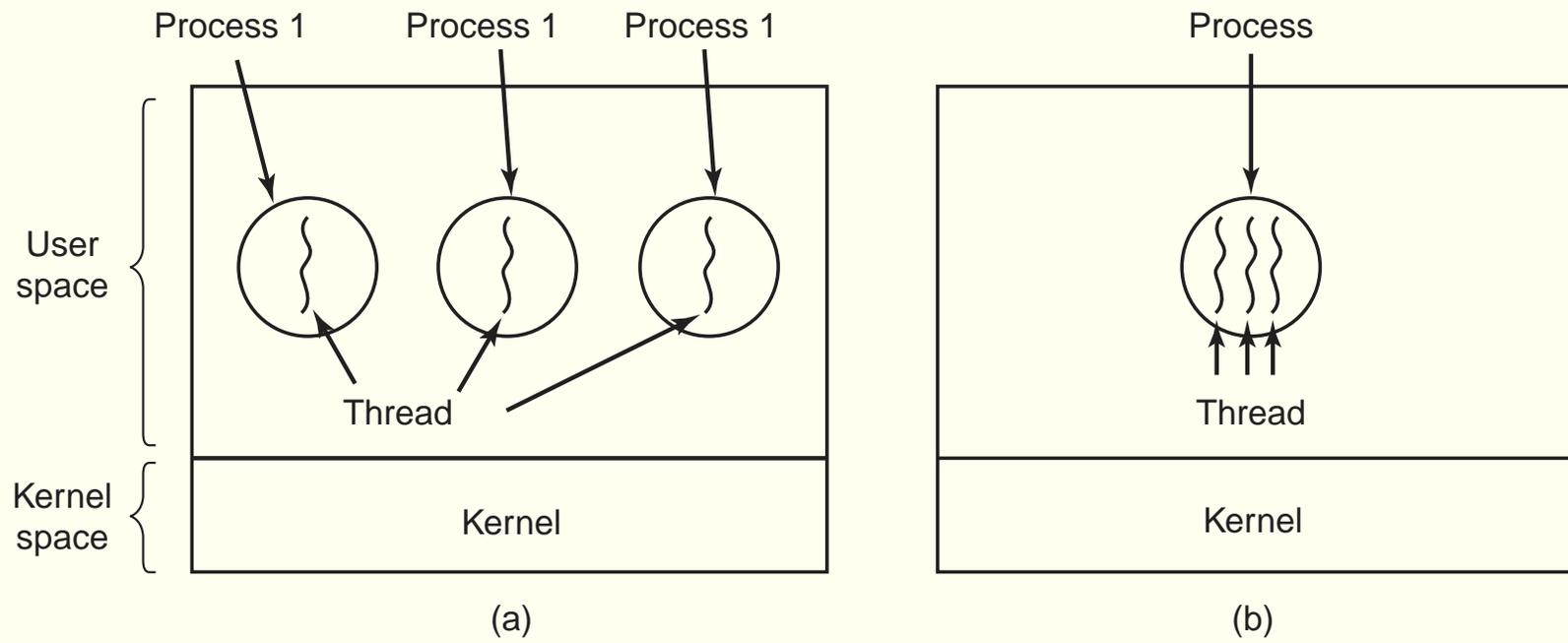
# Processo

- Programa em execução
- Espaço de endereçamento

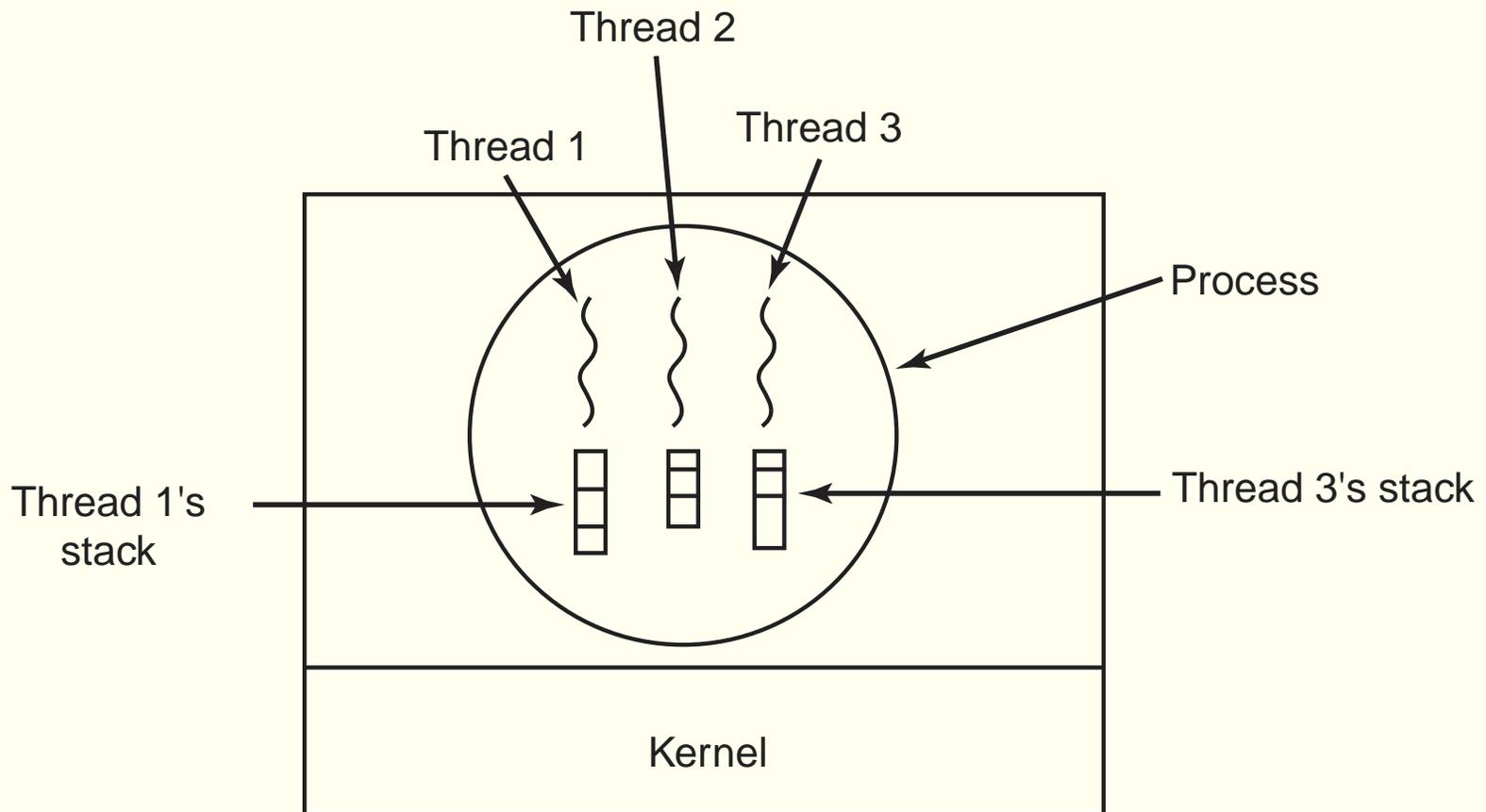


- Veja os códigos: `ender.c` e `ender-malloc.c`

# Modelo de threads



# Pilhas independentes



# Como trabalhar com threads

Veja os comandos:

- `pthread_create`
- `pthread_join`
- `pthread_exit`

## Como criar uma thread

```
int pthread_create(pthread_t *thread,  
                  pthread_attr_t *attr,  
                  void * (*start_routine)(void *),  
                  void *arg);
```

Veja o código: create0.c

## Como esperar por uma thread

```
int pthread_join(pthread_t th,  
                 void **thread_return);
```

Veja os códigos: join0.c, join1.c, join2.c, join3.c e join4.c

# Como passar parâmetros para uma thread

- Exemplo: cada thread pode precisar de um identificador único.
- Veja os códigos: `create1.c`, `create2.c`, `create3.c` e `create4.c`

# Como encerrar a execução de uma thread

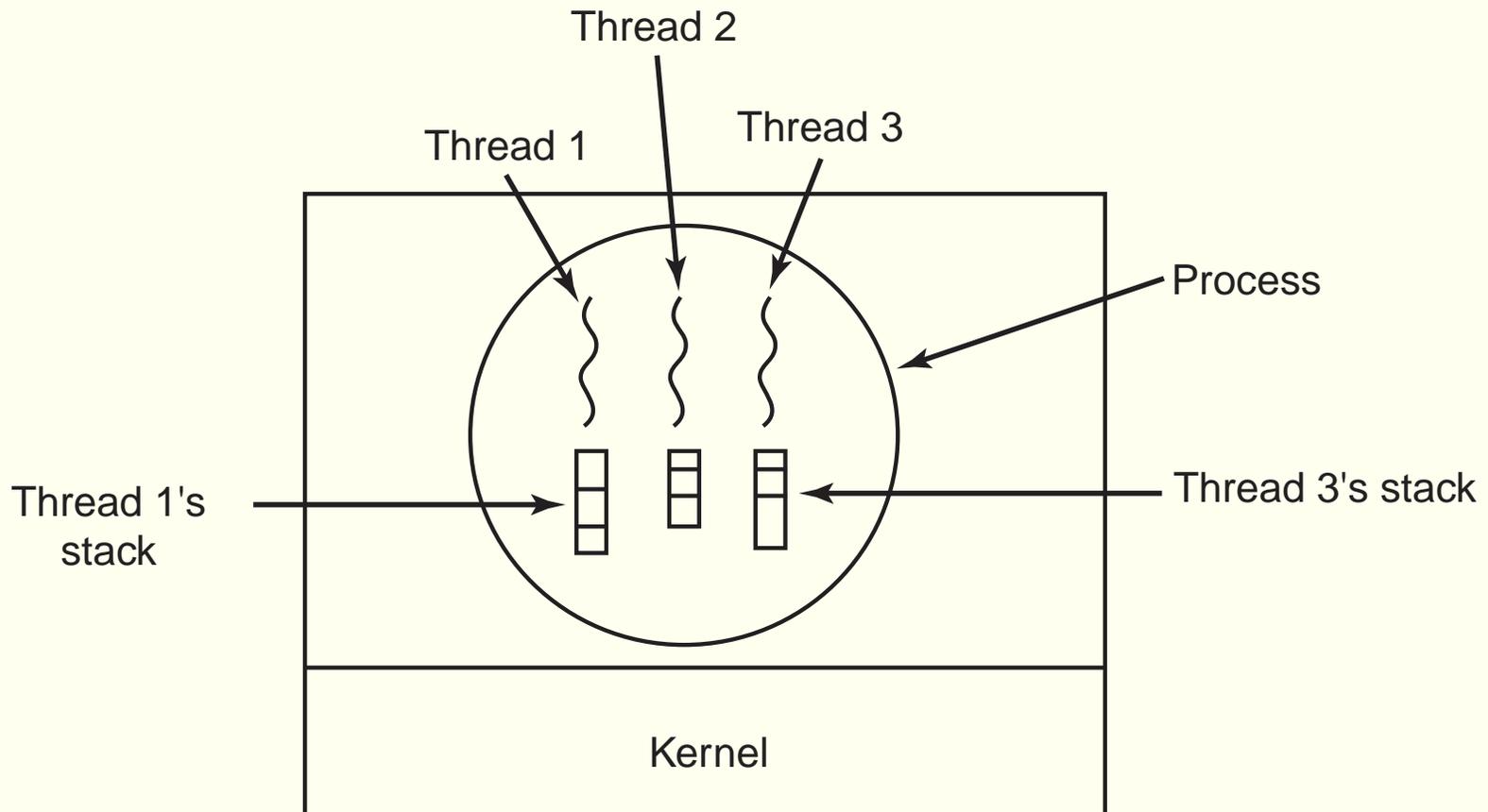
- Comando `return` na função principal da thread (passada como parâmetro em `pthread_create`)
- Análoga ao comando `return` na função `main()`

# Como encerrar a execução de uma thread

- `void pthread_exit(void *retval);`
- Análoga ao comando `exit(status);`

Veja o código: `exit0.c`

# Pilhas independentes



Analise o código pilha0.c para ver os endereços das pilhas.

## Exemplo de endereços

	⋮
0x804968C	global
	⋮
0x80483C4	main
0x8048378	f

## Exemplo de endereços

0xBFE7CB94	local_main
	⋮
0xBFE7CB70	param_f
	⋮
0xBFE7CB64	v[1]
0xBFE7CB60	v[0]
0xBFFFF680	⋮

# O que é armazenado na pilha?

- Espaço para valor de retorno da função (?)
- Argumentos
- Endereço de retorno
- Registradores
- Variáveis locais

## É muito fácil corromper a pilha

- Basta fazer acesso a posições não alocadas de um vetor;
- Veja o código: `corrompe_pilha.c` e `corrompe_pilha2.c`

# Uma thread pode corromper a pilha de outra thread

- Pilhas são independentes, mas não protegidas
- Veja o código: `corrompe_thread.c`