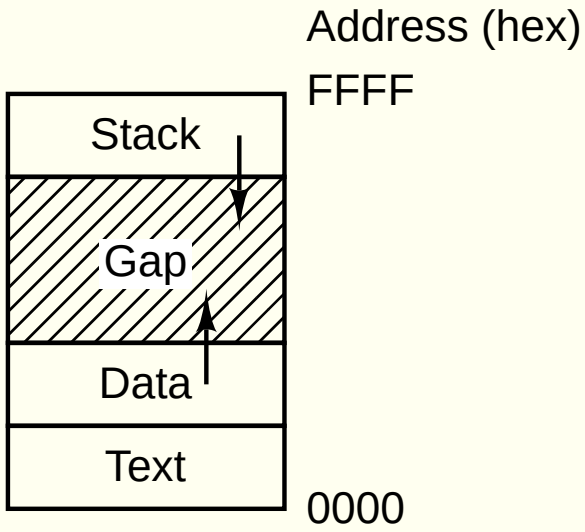
Processos e Threads

Aula 1

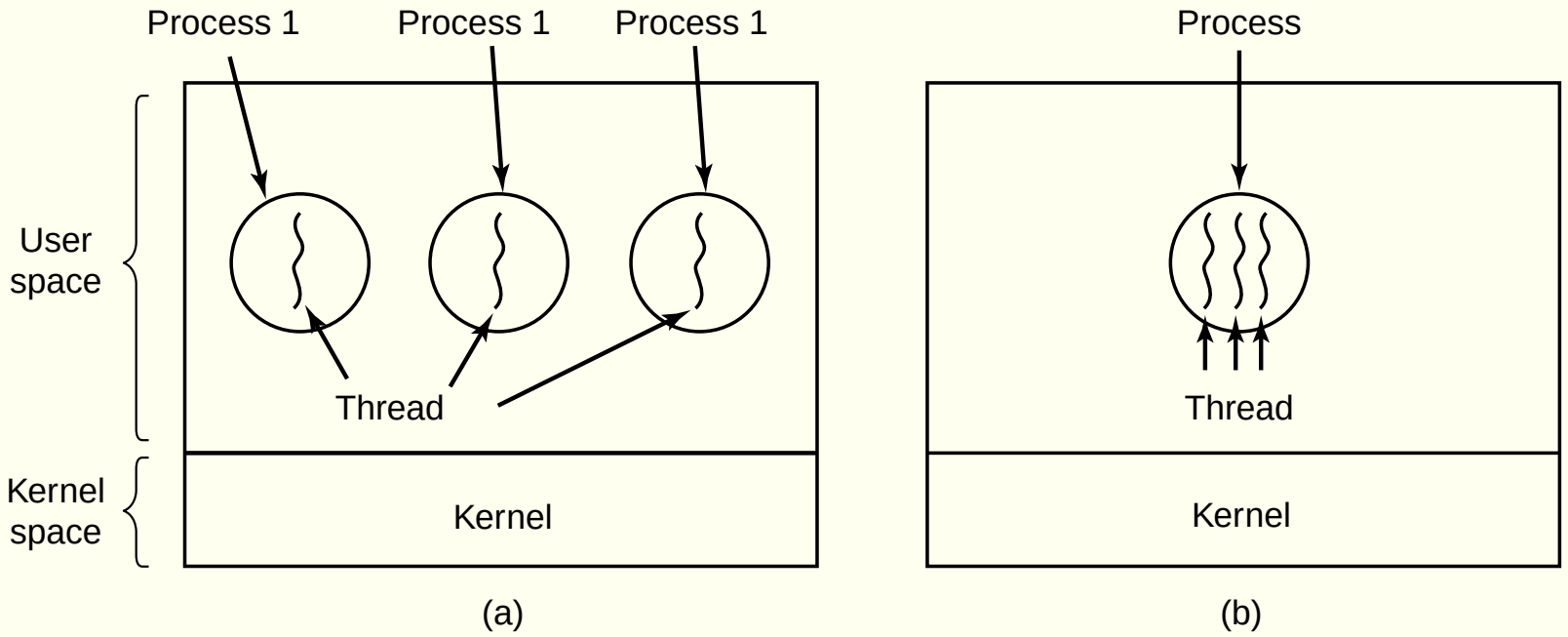
Processo

- Programa em execução
- Espaço de endereçamento

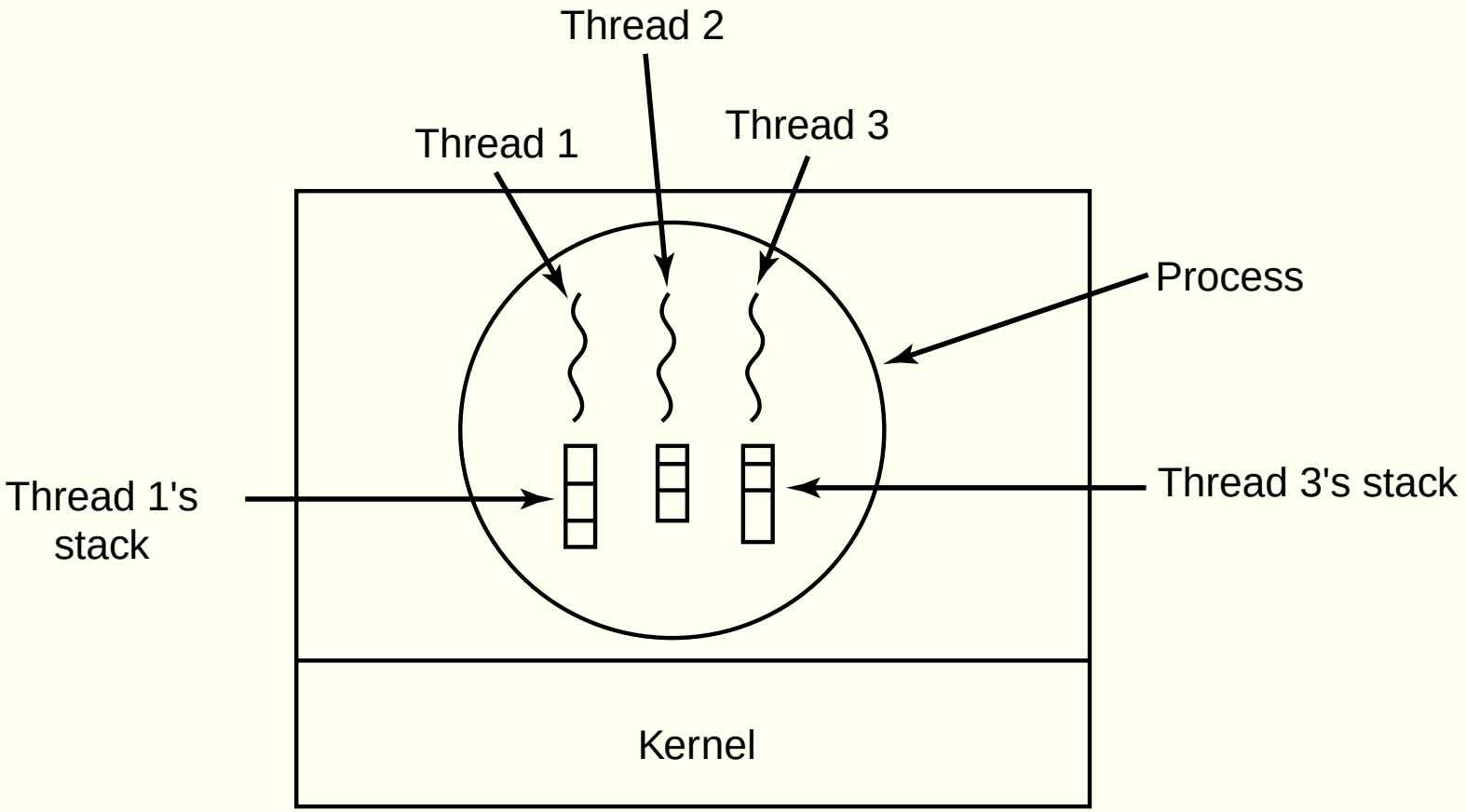


- Veja os códigos: ender.c e ender-malloc.c

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: create1.c, create2.c, create3.c e create4.c

Como passar parâmetros para uma thread

- Exemplo: cada thread pode precisar de um identificador único.
- Veja os códigos: create5.c e create6.c

Como encerrar a execução de uma thread

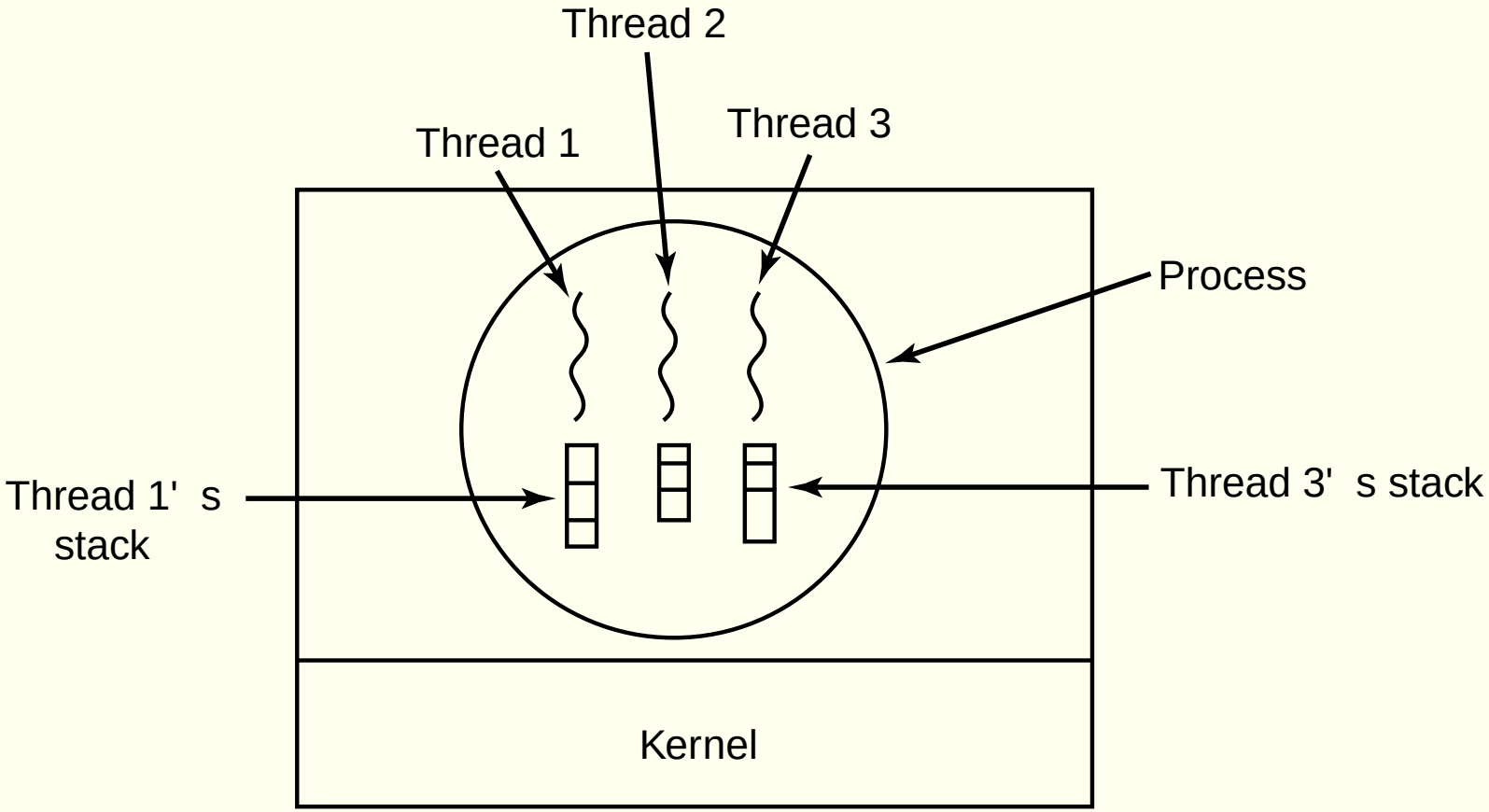
- Comando `return` na função principal da thread (passada como parâmetro em `pthread_create`)
- Análoga ao comando `return` na função `main()`

Como encerrar a execução de uma thread

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

Veja o código: create7.c

Pilhas independentes



Analise o código create8.c para ver os endereços das pilhas.

Laboratório 0

Multiplicação de matrizes

$C = A \times B$

- As colunas de C devem ser calculadas concorrentemente
- O formato para leitura/exibição das matrizes é:

```
3 4
0 1 2 3
0 2 3 4
1 2 7 8
```