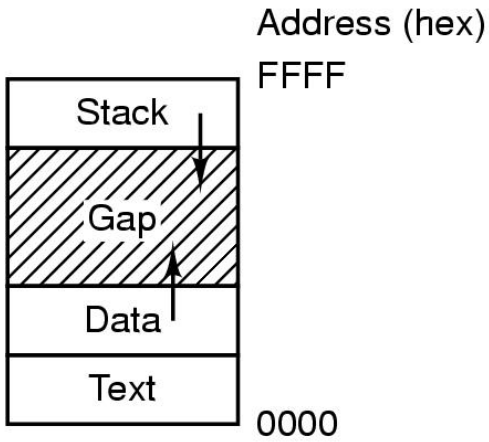
Sumário

Objetivos

- Processo
  - Espaço de endereçamento
- Pthreads
  - Operações create, join e exit

Processo

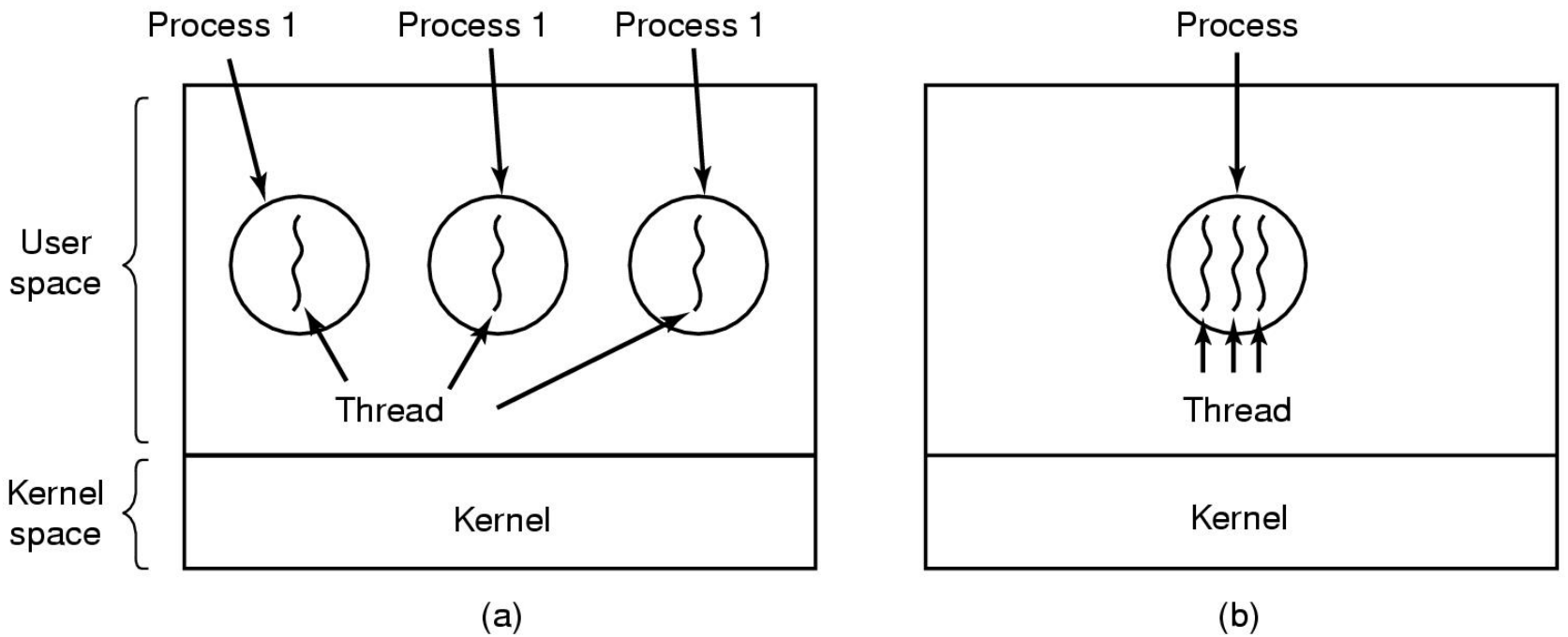
- Programa em execução
- Espaço de endereçamento



Tanenbaum: Figura 1.20

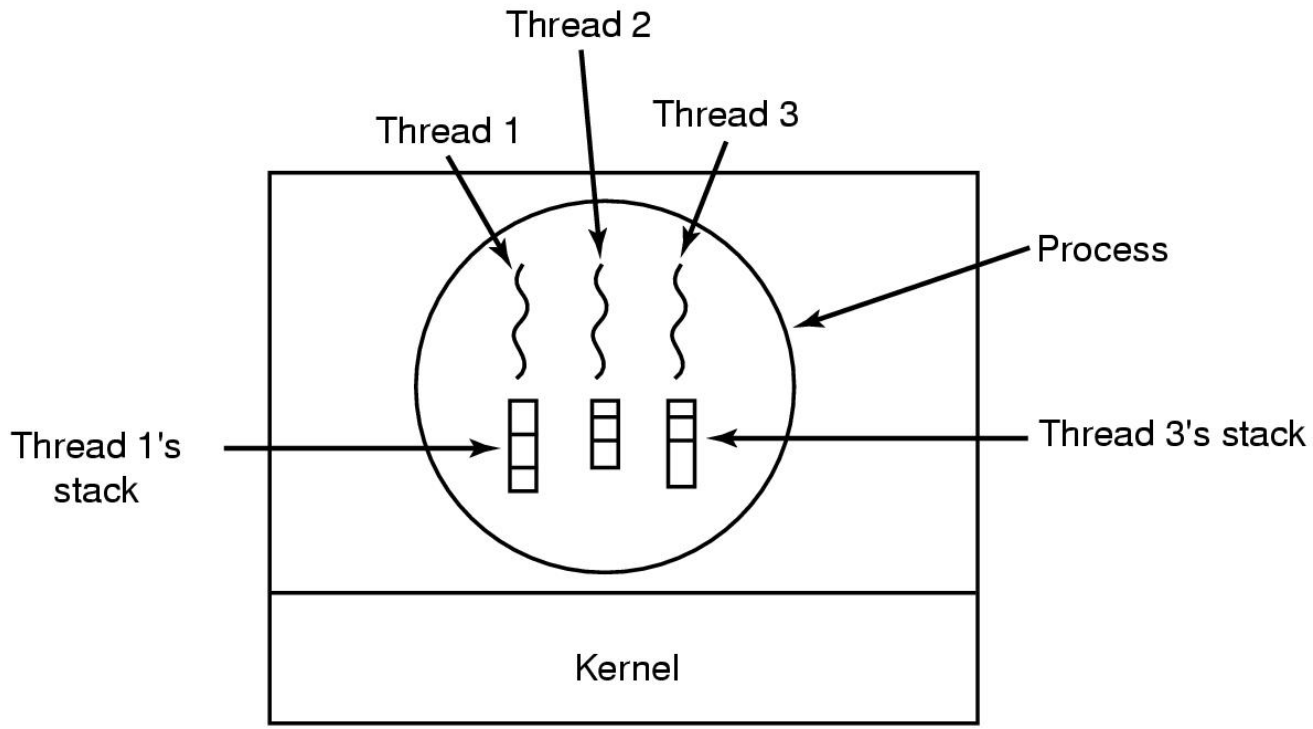
- Veja os códigos: ender.c e ender-malloc.c

Modelo de threads



Tanenbaum: Figura 2.6

Pilhas independentes



Tanenbaum: Figura 2.8

Como trabalhar com threads

Veja os comandos:

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

Para mais informações: `man <comando>`

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 thr,
                 void **thread_return);
```

Veja os códigos: join0.c

Como passar argumentos 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`)
- Análogo ao comando `return` na função `main()`

Veja os códigos: return0.c, return1.c pthread_return.c

Como encerrar a execução de uma thread

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

Veja os códigos: exit0.c, exit1.c e pthread_exit0.c