

PRÁTICAS EM PROGRAMAÇÃO

Trabalho 2

Mapeamento de Tarefas

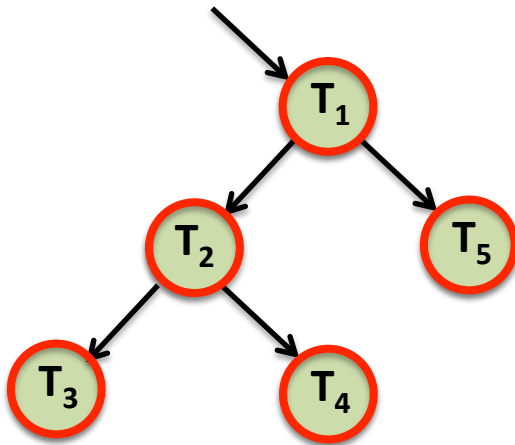
Prof. Ewerson Carvalho

ewerson.carvalho@unipampa.edu.br

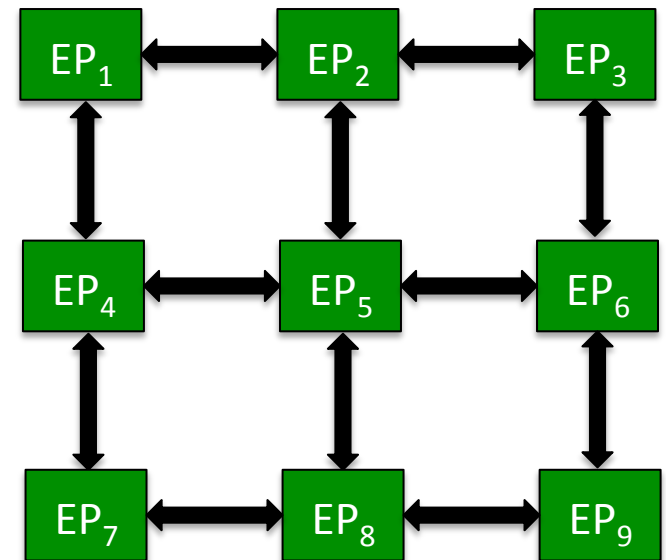
Introdução

MAPEAMENTO de Tarefas

DECIDE a **POSIÇÃO** de cada **TAREFA** em um **MPSoC**
VISA otimizar desempenho das Aplicações



Aplicação
Conjunto de Tarefas



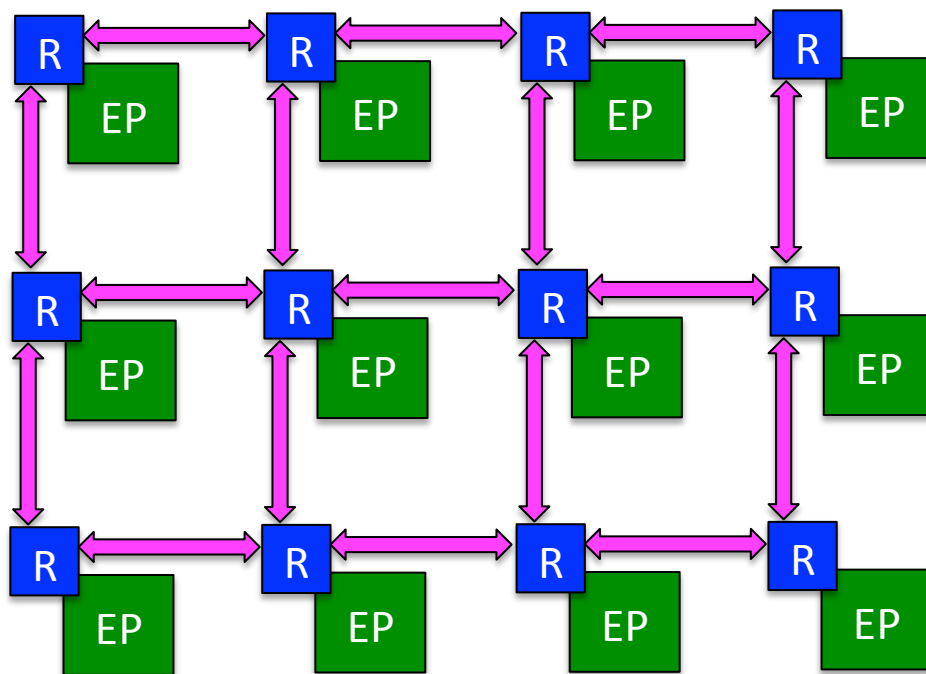
MPSoC
Conjunto de Elementos
de Processamento

MPSoC

SISTEMA MULTIPROCESSADO em um único CI

Composição:

- NoC = **ROTEADORES** + **CANAIS de Comunicação**
- EPs = **ELEMENTOS DE PROCESSAMENTO**



Variações:

- Tipos de EPs
- Topologia
- Roteamento
- Arbitragem
- Bufferização
- Tipo de Canal

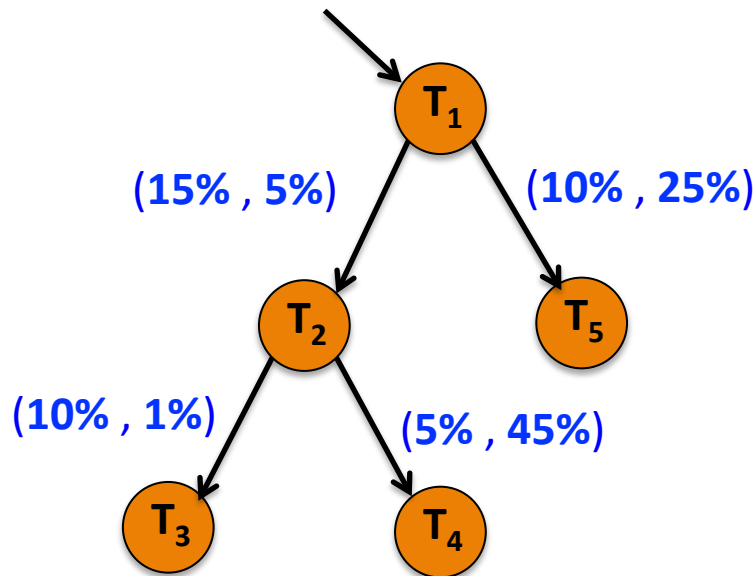
Aplicação

Conjunto de Tarefas (ou threads)

Representada por um **GRAFO**:

Vértices = **TAREFAS da APLICAÇÃO**

Arestas = **COMUNICAÇÕES** (em ambos os sentidos)

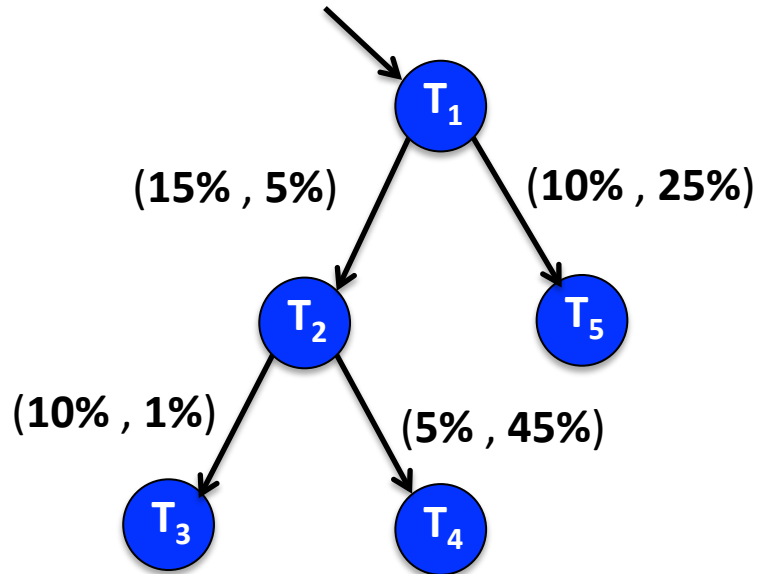


VALORES =
Taxas de Comunicação
(% ocupação do canal)
Ambos os sentidos

PROBLEMA

Mapeamento

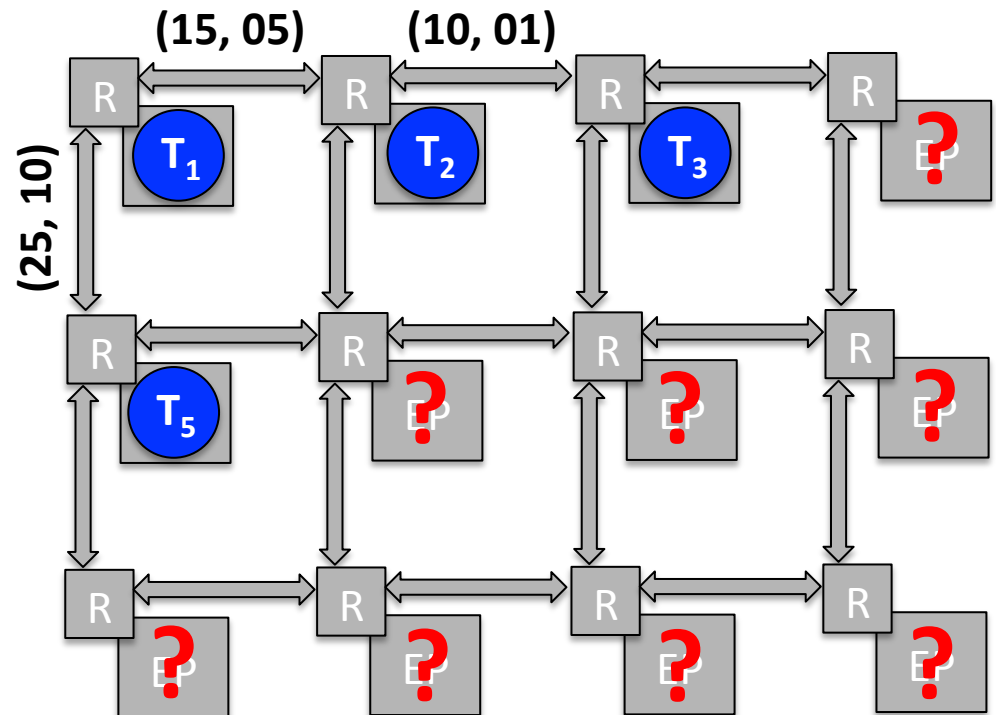
Descobrir a melhor posição para cada Tarefa



Adotando ROTEAMENTO XY

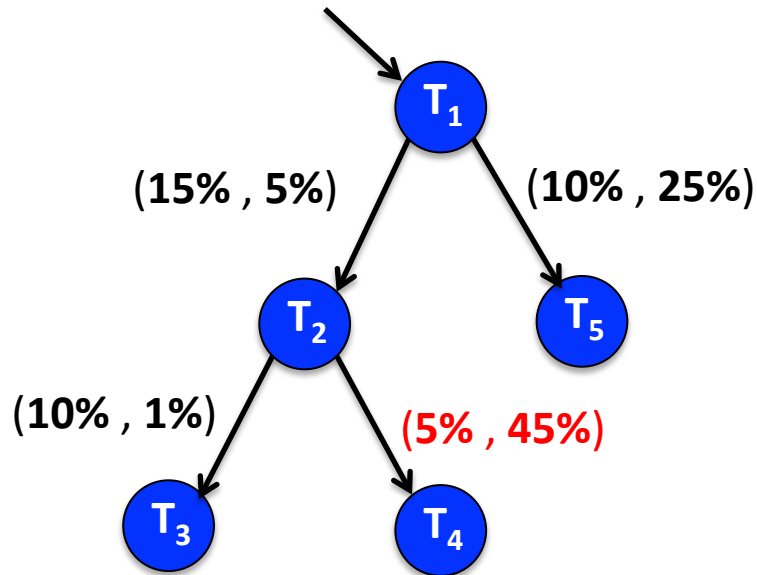
- Primeiro anda em X
- Depois anda em Y

T₄ Onde ?



Mapeamento

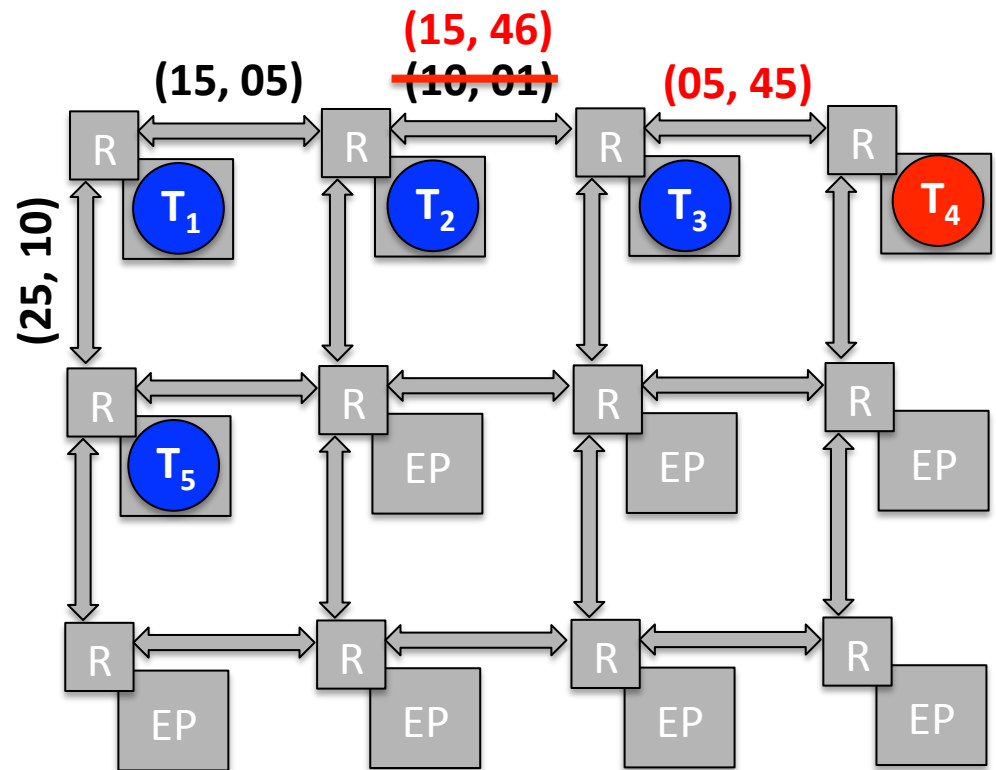
Descobrir a melhor posição para cada Tarefa



Adotando ROTEAMENTO XY

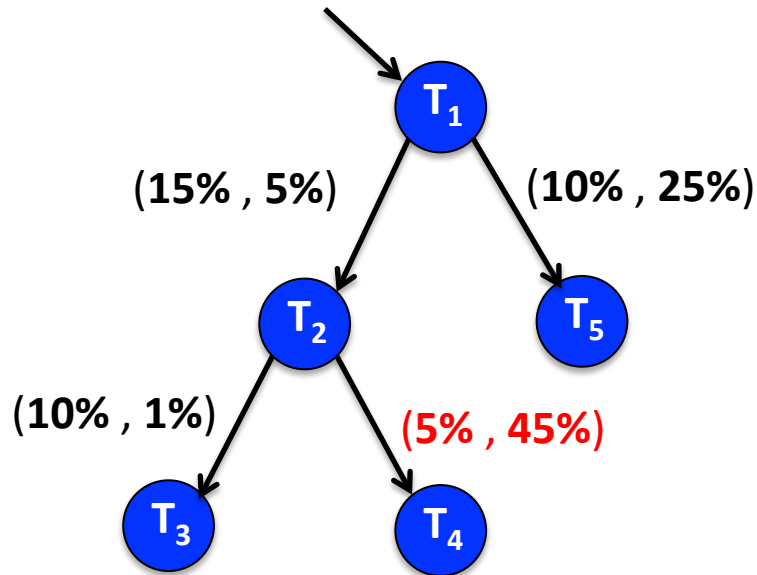
- Primeiro anda em X
- Depois anda em Y

Alternativa 1



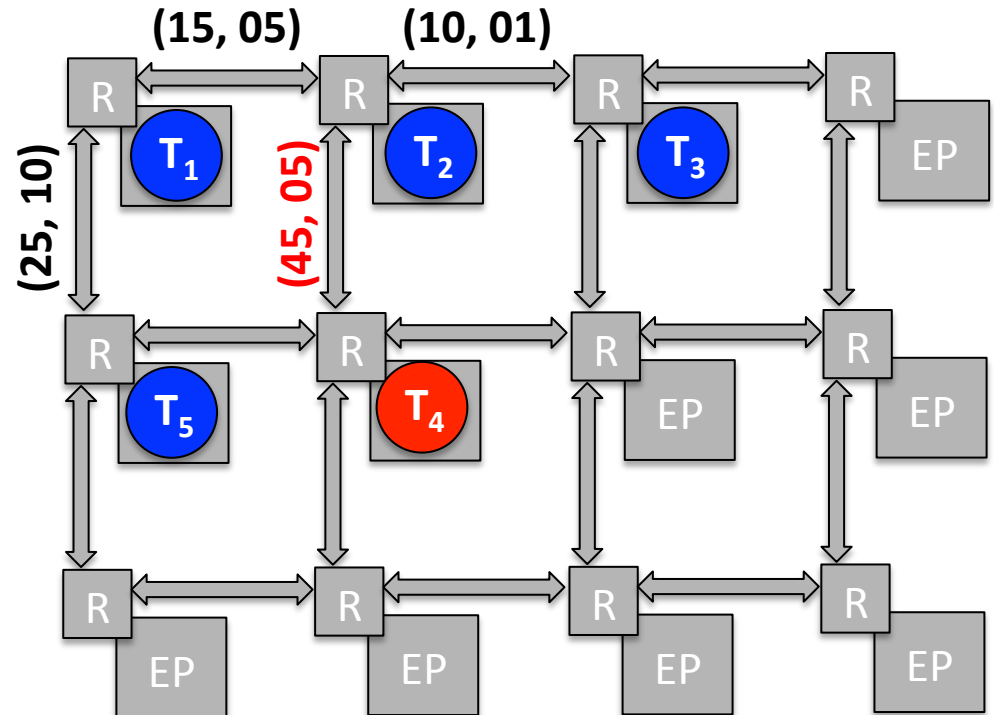
Mapeamento

Descobrir a melhor posição para cada Tarefa



Problema:
Várias alternativas!
Qual é a melhor?

Alternativa 2



**O QUE
IMPLEMENTAR?**

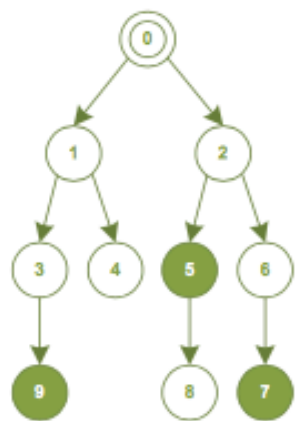
Implementar um Software

SIMULADOR DE MAPEAMENTO DE TAREFAS

Formato da ENTRADA

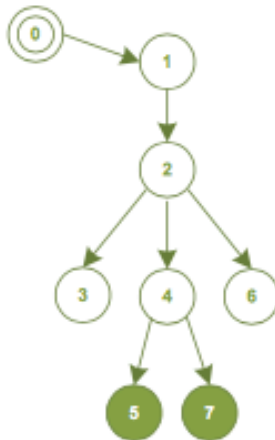
Arquivos com GRAFOs de APLICAÇÕES

Atenção:
Não é obrigatório
usar os Volumes!



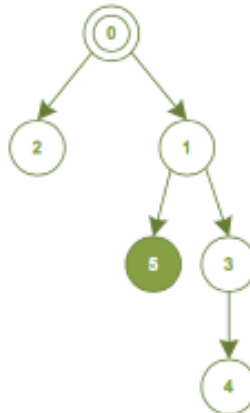
App0

0 [2, 150, 22%, 170, 20%]
0 [1, 340, 10%, 300, 15%]
1 [3, 470, 10%, 410, 20%]
1 [4, 300, 15%, 480, 15%]
2 [5, 380, 10%, 230, 10%]
2 [6, 240, 5%, 140, 5%]
3 [9, 210, 15%, 280, 25%]
5 [8, 190, 25%, 340, 20%]
6 [7, 120, 20%, 360, 25%]



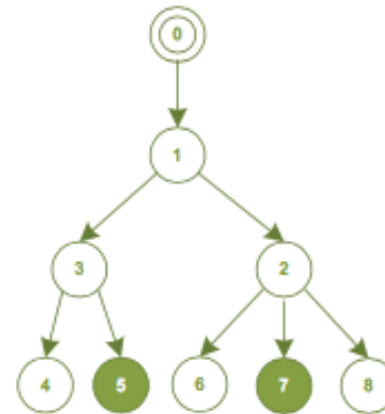
App1

0 [1, 260, 25%, 230, 20%]
1 [2, 330, 20%, 280, 5%]
2 [3, 310, 10%, 190, 5%]
2 [4, 190, 10%, 200, 5%]
2 [6, 160, 5%, 220, 15%]
4 [5, 190, 10%, 200, 10%]
4 [7, 260, 15%, 300, 15%]



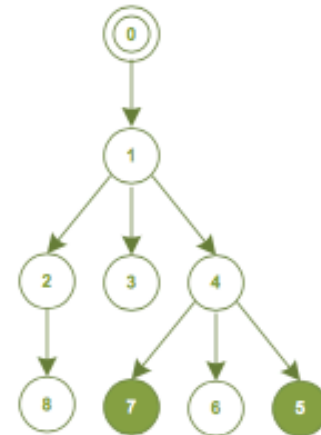
App2

0 [1, 240, 20%, 250, 10%]
0 [2, 260, 15%, 230, 20%]
1 [3, 330, 25%, 320, 10%]
1 [5, 260, 5%, 280, 5%]
3 [4, 260, 30%, 300, 15%]



App3

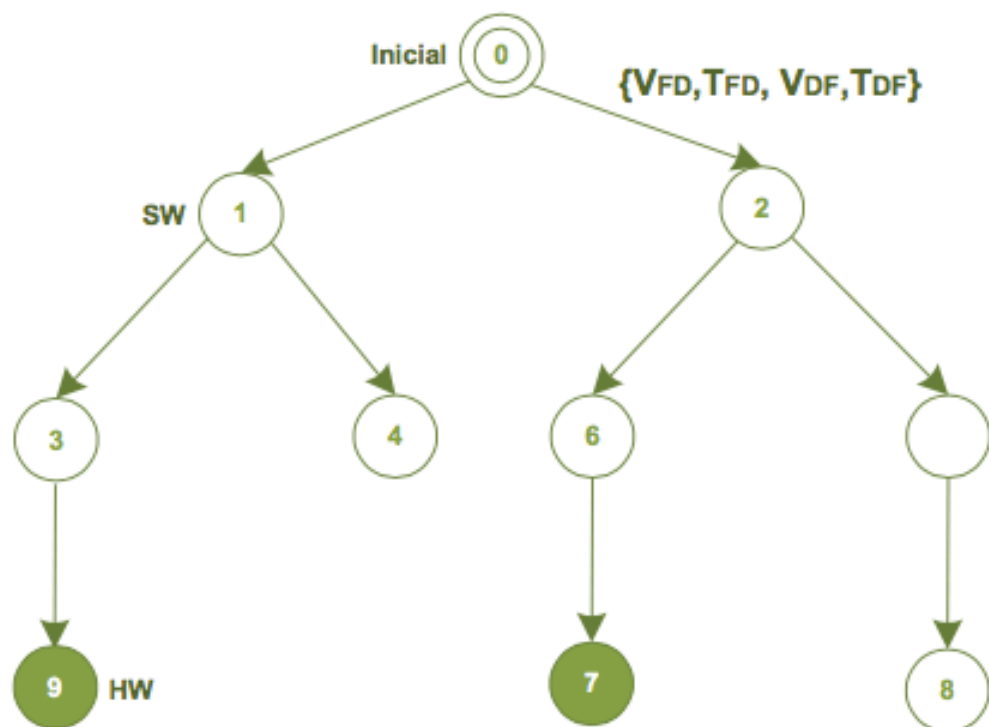
0 [1, 260, 20%, 260, 10%]
1 [2, 200, 15%, 170, 5%]
1 [3, 330, 10%, 320, 10%]
3 [4, 180, 15%, 170, 10%]
3 [5, 190, 5%, 160, 15%]
2 [6, 260, 15%, 230, 15%]
2 [7, 320, 5%, 320, 5%]
2 [8, 310, 5%, 190, 5%]



App4

0 [1, 240, 25%, 240, 10%]
1 [2, 190, 15%, 200, 5%]
1 [3, 180, 5%, 180, 5%]
1 [4, 240, 5%, 250, 5%]
4 [5, 310, 10%, 190, 6%]
4 [6, 200, 15%, 270, 10%]
4 [7, 200, 5%, 170, 10%]
2 [8, 260, 30%, 300, 20%]

Atenção:
Não é obrigatório
usar diferentes EPs!
 (ex.: SW e HW)



TREE 5%

0 [2, 1000, 5%, 10, 5%]
 0 [1, 1000, 5%, 10, 5%]
 1 [3, 1000, 5%, 10, 5%]
 1 [4, 1000, 5%, 10, 5%]
 2 [5, 1000, 5%, 10, 5%]
 2 [6, 1000, 5%, 10, 5%]
 3 [9, 1000, 5%, 10, 5%]
 5 [8, 1000, 5%, 10, 5%]
 6 [7, 1000, 5%, 10, 5%]

TREE 10%

0 [2, 1000, 10%, 10, 5%]
 0 [1, 1000, 10%, 10, 5%]
 1 [3, 1000, 10%, 10, 5%]
 1 [4, 1000, 10%, 10, 5%]
 2 [5, 1000, 10%, 10, 5%]
 2 [6, 1000, 10%, 10, 5%]
 3 [9, 1000, 10%, 10, 5%]
 5 [8, 1000, 10%, 10, 5%]
 6 [7, 1000, 10%, 10, 5%]

TREE 15%

0 [2, 1000, 15%, 10, 5%]
 0 [1, 1000, 15%, 10, 5%]
 1 [3, 1000, 15%, 10, 5%]
 1 [4, 1000, 15%, 10, 5%]
 2 [5, 1000, 15%, 10, 5%]
 2 [6, 1000, 15%, 10, 5%]
 3 [9, 1000, 15%, 10, 5%]
 5 [8, 1000, 15%, 10, 5%]
 6 [7, 1000, 15%, 10, 5%]

TREE 20%

0 [2, 1000, 20%, 10, 5%]
 0 [1, 1000, 20%, 10, 5%]
 1 [3, 1000, 20%, 10, 5%]
 1 [4, 1000, 20%, 10, 5%]
 2 [5, 1000, 20%, 10, 5%]
 2 [6, 1000, 20%, 10, 5%]
 3 [9, 1000, 20%, 10, 5%]
 5 [8, 1000, 20%, 10, 5%]
 6 [7, 1000, 20%, 10, 5%]



PIPE 5%

```

0 [ 1, 2000, 5%, 10, 5%]
1 [ 2, 2000, 5%, 10, 5%]
2 [ 3, 2000, 5%, 10, 5%]
3 [ 4, 2000, 5%, 10, 5%]
4 [ 5, 2000, 5%, 10, 5%]
5 [ 6, 2000, 5%, 10, 5%]
6 [ 7, 2000, 5%, 10, 5%]
7 [ 8, 2000, 5%, 10, 5%]
8 [ 9, 2000, 5%, 10, 5%]
  
```

PIPE 10%

```

0 [ 1, 2000, 10%, 10, 5%]
1 [ 2, 2000, 10%, 10, 5%]
2 [ 3, 2000, 10%, 10, 5%]
3 [ 4, 2000, 10%, 10, 5%]
4 [ 5, 2000, 10%, 10, 5%]
5 [ 6, 2000, 10%, 10, 5%]
6 [ 7, 2000, 10%, 10, 5%]
7 [ 8, 2000, 10%, 10, 5%]
8 [ 9, 2000, 10%, 10, 5%]
  
```

PIPE 15%

```

0 [ 1, 2000, 15%, 10, 5%]
1 [ 2, 2000, 15%, 10, 5%]
2 [ 3, 2000, 15%, 10, 5%]
3 [ 4, 2000, 15%, 10, 5%]
4 [ 5, 2000, 15%, 10, 5%]
5 [ 6, 2000, 15%, 10, 5%]
6 [ 7, 2000, 15%, 10, 5%]
7 [ 8, 2000, 15%, 10, 5%]
8 [ 9, 2000, 15%, 10, 5%]
  
```

PIPE 20%

```

0 [ 1, 2000, 20%, 10, 5%]
1 [ 2, 2000, 20%, 10, 5%]
2 [ 3, 2000, 20%, 10, 5%]
3 [ 4, 2000, 20%, 10, 5%]
4 [ 5, 2000, 20%, 10, 5%]
5 [ 6, 2000, 20%, 10, 5%]
6 [ 7, 2000, 20%, 10, 5%]
7 [ 8, 2000, 20%, 10, 5%]
8 [ 9, 2000, 20%, 10, 5%]
  
```

PIPE 25%

```

0 [ 1, 2000, 25%, 10, 5%]
1 [ 2, 2000, 25%, 10, 5%]
2 [ 3, 2000, 25%, 10, 5%]
3 [ 4, 2000, 25%, 10, 5%]
4 [ 5, 2000, 25%, 10, 5%]
5 [ 6, 2000, 25%, 10, 5%]
6 [ 7, 2000, 25%, 10, 5%]
7 [ 8, 2000, 25%, 10, 5%]
8 [ 9, 2000, 25%, 10, 5%]
  
```

PIPE 30%

```

0 [ 1, 2000, 30%, 10, 5%]
1 [ 2, 2000, 30%, 10, 5%]
2 [ 3, 2000, 30%, 10, 5%]
3 [ 4, 2000, 30%, 10, 5%]
4 [ 5, 2000, 30%, 10, 5%]
5 [ 6, 2000, 30%, 10, 5%]
6 [ 7, 2000, 30%, 10, 5%]
7 [ 8, 2000, 30%, 10, 5%]
8 [ 9, 2000, 30%, 10, 5%]
  
```

Formato da SAÍDA

VISUALIZAÇÃO **DINÂMICA** da OCUPAÇÃO do MPSoC

- OCUPAÇÃO dos ELEMENTOS de PROCESSAMENTO
- OCUPAÇÃO dos CANAIS da NOC

Implementar **3 POLÍTICAS** diferentes

Objetivo: Reduzir a carga nos Canais da NoC !!!!

- Política 1 - **FIRST FREE** (para comparação)
- Política 2 – **Proposta por VCs** (Otimização 1)
- Política 3 – **Proposta por VCs** (Otimização 2)

Atenção: Mapeamento é um problema NP-Completo!

Características MPSoC

MPSoC 6x6 (melhor se for configurável)

Topologia MALHA

Roteamento de Pacotes XY

Cenários para Teste

3 CENÁRIOS de Teste

Cada CENÁRIO com **5 APLICAÇÕES**

Cada APLICAÇÃO com **5 a 10 TAREFAS**

Avaliação

Formação do GRUPOS

4 ou 5 alunos

Obs.: Nem menos e nem mais.

Priorizar Mistura! (turma, curso, semestre)

**Eu não aceitarei
os trabalhos de
grupos fora das
especificações**

DATAS importantes

4 aulas para desenvolvimento

Entrega **Relatório: 13/09**

25/08 sex	07	Trabalho 2 - Enunciado e Fundamentação Teórica
30/08 qua	08	Trabalho 2 - Acompanhamento
01/09 sex	09	Trabalho 2 - Acompanhamento
06/09 qua	10	Trabalho 2 - Acompanhamento
08/09 sex		Feriado Estendido: Independência do Brasil
13/09 qua	11	Trabalho 2 - Entrega

Critérios de AVALIAÇÃO

- Completude do Trabalho
- Corretude dos Algoritmos
- Formatos de Entrada e Saída
- Estratégia de Armazenamento
- Estratégia de Simulação
- Qualidade do Relatório

O que ENTREGAR?

- **RELATÓRIO** (.pdf)
- **Fontes COMPACTADOS** (.rar, .zip., .7z)

Enviar arquivos para:

ewerson.carvalho@unipampa.edu.br

Estrutura do RELATÓRIO

1. INTRODUÇÃO

Breve introdução do trabalho.

Explicação do que será implementado.

Como funcionam os algoritmos a serem implementados?

2. DESENVOLVIMENTO

Explicação dos códigos desenvolvidos.

Apenas as partes relevantes dos códigos.

Justifiquem suas escolhas (linguagem, estrutura de dados, interface etc).

Use subseções para cada algoritmo implementado.

3. RESULTADOS OBTIDOS

Primeiramente, deve-se **explicar os cenários de teste** utilizados.

Cada resultado obtido deve ser **apresentado e discutido.**

Use subseções para cada cenário/resultado.

Tem que mostrar que os algoritmos funcionaram.

Ao final, **qual foi o melhor resultado, e o porquê?**

Aproveitem as observações do relatório passado para fazer um Relatório Melhor!

4. CONCLUSÃO

Considerações sobre o trabalho desenvolvido.

Discussão sobre **experiências adquiridas** e **dificuldades encontradas.**

Bom Trabalho!