

UNIVERSIDADE DE SÃO PAULO (EACH)

ACH2096 - LABORATÓRIO DE SISTEMAS  
OPERACIONAIS

# Manual de Compilação e Execução

*Matheus Preischadt Pinheiro*  
*8623799*

supervisionado por  
Dr. Gisele DA SILVA CRAVEIRO

30 de março de 2016

## 1 Informações de Compilação

O programa pode ser compilado em UNIX ou Windows através do programa Make e do pacote GCC (MinGW basic para Windows). Como a quantidade de linhas de código é pequena, todos os arquivos precisam ser recompilados em caso de alteração. Dessa forma, a compilação deve ser feita da seguinte maneira:

```
make clean
make
```

**OBS:** Deve existir um diretório "bin" na pasta onde se encontra o arquivo MakeFile, caso contrário, o executável não será criado!

## 2 Informações de Entrada

A entrada pode ser inserida através do teclado ou por um arquivo em formato *txt* e é interpretada da seguinte maneira: a primeira linha da entrada deve ter **cinco inteiros** separados por **espaços**. Cada um desses inteiros corresponde, respectivamente, a **tamanho da memória principal**, **tamanho dos quadros**, **CPU time slice do Round-Robin**, **duração das operações de disco** e **número de jobs que serão processados**. As linhas seguintes correspondem aos processos que serão executadas e também deverão conter *cinco inteiros*, sendo esses: **ID do processo**, **tempo de chegada**, **CPU-burst**, **quantidade de memória** e **quantidade de requests IO**.

```
100 30 20 3 4
1 1 11 700 3
2 1 15 25 5
3 9 18 500 2
4 3 12 1650 3
```

## 3 Informações de Execução e Saída

A execução *crua* do programa pode ser feita da seguinte maneira:

**Windows:**  
*"bin/BASYS"*

### UNIX:

`./bin/BASYS`

Dessa maneira, o programa será executado e esperará pela entrada do teclado.

Para utilizar a entrada de texto, os seguintes comandos devem ser utilizados:

### Windows:

`"bin/BASYS" < 'nomeArquivoEntrada'.txt`

### UNIX:

`./bin/BASYS < 'nomeArquivoEntrada'.txt`

## 3.1 Saída

A saída do programa é gerada no próprio terminal e pode ser acompanhada dessa maneira. Entretanto, problemas com as diferenças entre *tabs*, espaços e afins em Windows e UNIX podem gerar problemas na formatação de saída. Assim, é recomendado utilizar-se do comando que gera um arquivo *txt* com a saída do programa. Tal pode ser feito da seguinte maneira:

### Windows:

`"bin/BASYS" < 'nomeArquivoEntrada'.txt > 'nomeArquivoSaida'.txt`

### UNIX:

`./bin/BASYS < 'nomeArquivoEntrada'.txt > 'nomeArquivoSaida'.txt`

Da maneira acima, o programa rodará utilizando o arquivo *nomeArquivoEntrada'.txt* como entrada e criando o arquivo *'nomeArquivoSaida'.txt* como saída. Como no exemplo a seguir:

### Entrada:

```
100 5 20 3 1
1 1 5 500 1
```

### Saída:

Memory Size	Quad Size	Round-Robin Time Slice	Disk Write Rate	Number of Jobs
100	5	20	3	1
Job ID	Next Event	Scheduled Time		
1	Job Arrival	1		
Job ID	Next Event	Scheduled Time		
1	Memory Request	1		
Job ID	Next Event	Scheduled Time		
1	CPU Request	1		
Job ID	Next Event	Scheduled Time		
1	CPU Release	6		
Job ID	Next Event	Scheduled Time		
1	IO Request	6		
Job ID	Next Event	Scheduled Time		
1	IO Release	9		
Job ID	Next Event	Scheduled Time		
1	CPU Request	9		
Job ID	Next Event	Scheduled Time		
1	Job Completion	14		

A saída mostra a tabela de eventos completa a cada iteração do LOOP principal do sistema.