Neste experimento foram realizadas 50 tentativas para cada tipo de granularidade. Para o teste parcialmente aleatório, foi modificada a propriedade `NUM_WORKERS` com valores em aberto e para o teste determinado foi alterada esta mesma propriedade com valores de 1 a 60.

A diferença entre as tentativas totalmente aleatórias e as outras duas granularidades foi grande. Este resultado já era esperado, visto que a depuração de uma aplicação sem informação nenhuma à priori tem a sua efetividade ligada à probabilidade de encontrar tanto a falha quanto a sua causa.

Entretanto não houve muita alteração entre os tipos determinado e parcialmente aleatório. Isto ocorreu devido à limitação na quantidade de propriedades e de seus possíveis valores de troca da aplicação, ou seja, com tal restrição as trocas com sucesso foram semelhantes nas duas configurações.

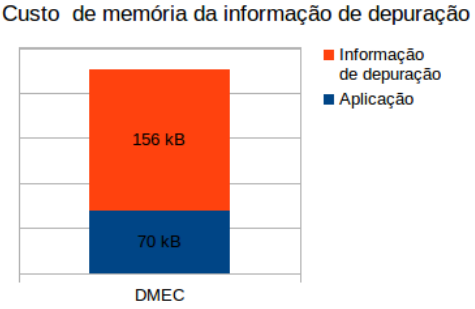

Figura 15 – Consumo de memória extra para armazenar as informações de depuração.

Conforme apresentado na Figura 15, a aplicação não tem uma imagem grande, mas quando adicionamos a informação extra em tempo de compilação, o consumo de memória foi aumentado em cerca de 200%. Em um sistema embarcado real, o tamanho desta nova imagem seria proibitivo.

6.2 EXPERIMENTO 2 - UTILIZAÇÃO EM DISCIPLINA E EFETIVIDADE DOS RESULTADOS

Este experimento foi elaborado para verificar a utilização da ferramenta como artifício para estudo de disciplinas que envolvem codificação de *software* embarcado. Além disso, este experimento tem como objetivo secundário a avaliação da ferramenta e da efetividade dos seus resultados.

Para verificar a viabilidade do uso em disciplinas o experimento foi aplicado em códigos fonte de desenvolvedores inexperientes na produção de

software embarcado. A ideia é entender como a ferramenta pode ser introduzida em disciplinas de desenvolvimento de *software* considerando a facilidade de uso, abrangência dos testes e ajuste fino do relatório gerado.

TAP foi avaliada através do código fonte produzido por estudantes do curso de bacharelado em ciências da computação e que estavam cursando a disciplina de sistemas operacionais II. A escolha da turma se deu devido ao fato de ser uma disciplina com muitos exercícios práticos, desenvolvidos no sistema operacional EPOS - cuja disponibilidade é controlada por senha, ou seja, sabe-se de antemão quais estudantes já tiveram contato com o sistema.

No início desta disciplina os alunos tem acesso a um EPOS incompleto e cada exercício realizado tem o intuito de implementar uma parte deste *software*. Todos os exercícios tem um escopo fechado e sabe-se quais testes devem ser executados corretamente para cada uma das fases.

O EPOS permite que cada aplicação gerada possua um arquivo próprio de configuração de abstrações para definir o seu comportamento. Dentre estas configurações encontram-se definições como, por exemplo, o tipo de escalonamento, que podem afetar um determinado comportamento da aplicação sem modificar seu propósito.

Se existem várias configurações válidas de uma mesma aplicação, todas elas devem ser atendidas pelas soluções dos exercícios. Entretanto, devido à inexperiência, alguns alunos não utilizam esta troca de parâmetros como ferramenta de apoio.

6.2.1 Configuração do experimento e execução dos testes

Para fornecer dicas sobre as configurações mais importantes, TAP foi configurada para realizar a troca das seguintes configurações: tipo de escalonamento de processos, o quantum de tempo e tamanhos da *stack* e *heap*. O arquivo de configuração desenvolvido para a disciplina pode ser observado na Figura 16.

A configuração da ferramenta para a disciplina focou em algumas configurações, mas com alta variabilidade de valores candidatos para a substituição. TAP foi configurada desta forma para que seja possível apresentar aos alunos a quantidade de aplicações que devem funcionar corretamente após a implementação de um determinado exercício.

Quando o foco é o aprendizado, é justo que esta configuração seja bem abrangente, visto que esta variedade pode ampliar a visão para cenários que talvez sejam importantes e que podem ter sido negligenciados durante a fase de concepção, incluindo a idealização dos requisitos não funcionais.

Inicialmente o EPOS conta com 42 aplicações de teste e em cada uma

```
<test>
   <configuration>
     <trait scope="application" id="STACK_SIZE">
         <value>512</value>
         <value>1024</value>
         <value>2048</value>
         <value>3072</value>
         <value>4096</value>
      <trait>

      <trait scope="application" id="HEAP_SIZE">
         <value>512</value>
         <value>1024</value>
         <value>2048</value>
         <value>3072</value>
         <value>4096</value>
      <trait>

      <trait scope="System" id="STACK_SIZE">
         <value>512</value>
         <value>1024</value>
         <value>2048</value>
         <value>3072</value>
         <value>4096</value>
      <trait>

      <trait scope="System" id="HEAP_SIZE">
         <value>512</value>
         <value>1024</value>
         <value>2048</value>
         <value>3072</value>
         <value>4096</value>
      <trait>

      <trait scope="System" id="QUANTUM">
         <value>10000</value>
         <value>20000</value>
         <value>50000</value>
         <value>100000</value>
         <value>200000</value>
         <value>500000</value>
      <trait>

     <trait scope="Thread" id="QUANTUM">
         <value>10000</value>
         <value>20000</value>
         <value>50000</value>
         <value>100000</value>
         <value>200000</value>
         <value>500000</value>
      <trait>

     <trait id="Scheduling_Criteria">
         <value>"RR"</value>
         <value>"FCFS"</value>
      <trait>
   </configuration>
<test>
```

Figura 16 – Configuração de TAP para as aplicações do EPOS da disciplina.

foram realizadas 16.200 trocas, ou seja, TAP trocou as configurações destas aplicações e gerou um total de 680.400 variações, as quais foram automaticamente executadas e depuradas. Para cada aplicação (e suas variações) foram necessárias cerca de 180 horas de processamento.

Esta abrangência no resultado gera uma grande quantidade de dados, portanto, para simplificar o estudo da execução dos testes, o relatório fornecido por TAP apresenta informações resumidas, com a opção de verificar os arquivos de *log* das execuções falhas. Na Figura 17 encontra-se um trecho de relatório gerado pela ferramenta.

```
= = = = = TEST REPORT = = = = =
= Configurations =
Quantum: 10ms, 20ms, 50ms, 100ms, 200ms, 500ms
Stack: 512MB, 1GB, 2GB,3GB 4GB
Heap: 512MB, 1GB, 2GB,3GB 4GB
Schedulers: RR, FCFS

= Application: active_test=
Successful tests:5160 - Failed tests:11040
Failed tests execution log and debugging information on attachment.
```

Figura 17 – Trecho de relatório gerado por TAP para as aplicações do EPOS da disciplina.

Os arquivos com as execuções falhas contém informações desde o momento de compilação até o relatório final da depuração com o GDB. Cada execução tem seu próprio arquivo, identificado pelo próprio nome concatenado com as configurações que foram trocadas.

Utilizando como exemplo a Figura 17, nota-se que a aplicação *active_test* continha algumas execuções falhas, sendo assim, dentre os anexos devemos encontrar 736 arquivos cujo nome iniciam-se com esta aplicação. Por exemplo, o arquivo *active_test_Scheduler_FCFS.log* é um relatório de falha da execução dos testes da aplicação *active_test* para quando a configuração *Scheduler* estava com o valor *FCFS*.

A Figura 18 apresenta um resumo das execuções de falha dos exercícios $E1$ a $E7$ através da solução dos grupos ($G1$ a $G12$). Nela é possível notar que o número de execuções com falha começa a diminuir conforme são entregues os exercícios.

Este resultado era esperado se for considerado o aumento da familiaridade com o sistema operacional. Outro fato crucial para este resultado está na realização incremental dos trabalhos, em que o aluno precisa ter a implementação do trabalho n para entregar corretamente o trabalho $n+1$.

Os casos de falha apresentados na Figura 18 pode-se perceber que houve uma a grande tendência de redução da quantidade de execuções com

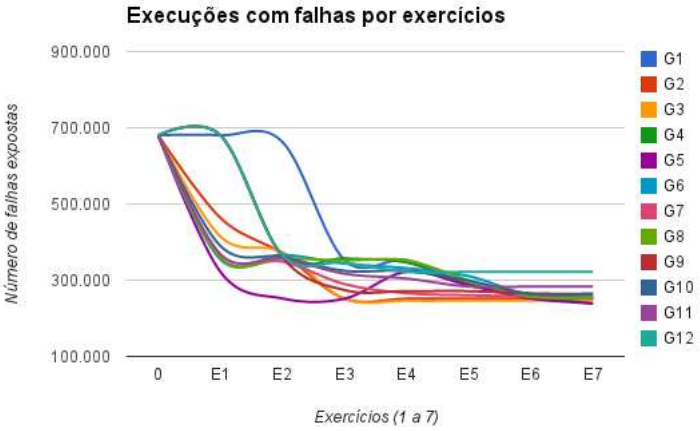

Figura 18 – Execuções dos exercícios que resultaram em falha.

falha no início da disciplina, que começou a estabilizar e seguir com mais uma leve redução. Isto ocorreu porque os exercícios de $E1$ até $E3$ relacionam-se diretamente com o assunto *threads*, criando uma certa zona de estabilidade. Esta estabilidade se quebrou à partir do $E4$, cuja avaliação envolve o conceitos como o de eventos programados e rotina de tratamento de interrupções.