

1 Workload 1

1.1 Descripcio del proces

El primer workload prova el rendiment de les diferents politiques de planificacio amb processos intensius en calcul, amb molt poca entrada/sortida. El proces pare crea un fill, que tambe crea un fill (3 procesos en total) i tots calculen el n-essim terme de la succecio de fibonacci.

1.2 Codi

```
1  int fib(int n)
2  {
3      if (n == 0) return 0;
4      if (n == 1) return 1;
5      return fib(n-1) + fib(n-2);
6  }
7
8  void workload1(){
9      int pid, pid_f, f;
10     pid = fork();
11     if(pid == 0){
12         pid_f = fork();
13         if(pid_f == 0)
14             f = fib(500);
15         else
16             f = fib(750);
17     }
18     else{
19         f = fib(1000);
20     }
21 }
```

2 Workload 2

2.1 Codi

```
1  void workload2(){
2      int pid, pid_f, f, r, i;
3      pid = fork();
4      if(pid == 0){
5          pid_f = fork();
6          if(pid_f == 0){
7              for(i = 0; i < 10; ++i)
```

```

8         r = read(0,&buff, 10);
9         write(1,"1\n",2);
10    }
11    else{
12        for(i = 0; i < 5; ++i)
13            r = read(0,&buff, 20);
14            write(1,"2\n",2);
15    }
16 }
17 else{
18     for(i = 0; i < 20; ++i)
19         r = read(0,&buff, 5);
20         write(1,"p\n",2);
21     }
22 }

```

3 Workload 3