

*Я в своем познании настолько преисполнился, что я как будто бы уже
сто триллионов миллиардов лет проживаю на триллионах и
триллионах таких же планет, как эта Земля, мне этот мир абсолютно
понятен, и я здесь ищу только одного - покоя, умиротворения и
вот этой гармонии...*

эксперименты с последовательным и параллельным выполнением вычислительно сложных задач

эксперимент 1

`lnn.sh` – вычисляет значение функции $\ln(1+x)$ для x в диапазоне $(-1, 1)$ по формуле маклорена до 50-ого слагаемого:

```

1 #!/bin/bash
2 #lnn(1+x)=sum((-1)^(n-1)x^n/n),n=1->inf
3 ww=0 xn=1 sgn=-1
4 for ((i=1;i<50;i++)); do
5   ((sgn=sgn*(-1)))
6   xn=$((echo "scale=30; $xn * $1" | bc)
7   ww=$((echo "scale=30; $sgn * $xn / $i + $ww" | bc)
8   done
9 echo $ww

```

`1b.sh` – для данного n последовательно считает значение функции в n точках:

```

1 #!/bin/bash
2 ((n=$1+1))
3 period=$((echo "scale=6; 2 / $n" | bc)
4 chmod +x lnn.sh
5 for ((i=1;i<$n;i++)); do
6   dot=$((echo "scale=6; -1 + $i * $period" | bc)
7   ./lnn.sh $dot
8 done

```

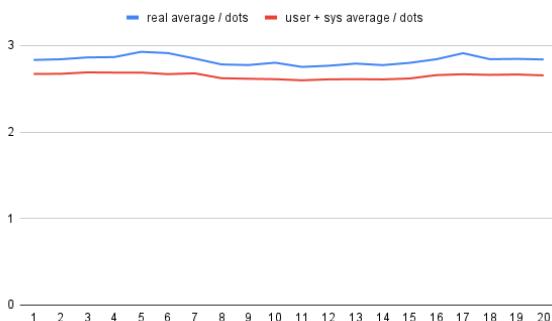
`1c.sh` – для каждого n от 1 до 20 запускает скрипт, считающий значение в n точках 10 раз, и выводит полученные значения:

```

1 #!/bin/bash
2 > report.1c
3 for ((i=1;i<21;i++)); do
4   echo $i >> report.1c
5   for ((j=1;j<11;j++)); do
6     { time ./1b.sh $i ; } 2>> report.1c
7   done
8   echo "" >> report.1c
9 done

```

средние арифметические значения времени вычисления
функции в точке в зависимости от n



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
real	2.793	5.079	6.304	11.465	14.830	17.450	20.479	21.705	25.185	29.526	29.920	32.195	36.864	36.176	42.594	45.106	48.956	52.942	63.105	59.024
user	1.766	3.497	5.128	7.084	8.765	10.402	12.222	13.545	15.511	17.185	18.920	20.377	22.234	23.811	25.928	27.519	29.539	31.511	32.555	34.293
sys	0.881	1.765	2.720	3.682	4.676	5.518	6.987	6.987	8.212	8.776	8.342	10.084	11.778	12.271	13.692	14.768	16.381	17.540	17.540	18.489
real	2.792	6.094	9.434	11.435	14.831	17.528	20.177	22.739	24.457	27.829	30.615	33.195	36.551	38.718	41.743	44.682	50.223	50.375	52.946	55.912
user	1.735	3.665	5.500	6.979	8.831	10.488	12.205	13.898	15.244	17.147	19.079	20.591	22.457	24.072	25.634	27.406	29.587	30.954	32.489	34.205
sys	0.905	2.017	3.231	3.770	4.746	5.591	6.748	7.509	7.873	8.655	9.769	10.670	11.945	12.431	13.714	14.444	16.787	16.516	16.945	18.503
real	2.839	5.686	8.506	11.517	14.686	17.598	20.438	21.859	24.798	28.267	30.228	32.633	36.746	39.261	42.730	45.018	50.391	51.577	54.435	55.831
user	1.749	3.481	5.148	7.021	8.812	10.548	12.250	13.625	15.538	17.097	18.883	20.490	22.463	24.115	25.787	27.536	29.756	30.921	32.750	34.485
sys	0.929	1.859	2.857	3.812	4.675	5.613	6.920	7.025	7.876	9.096	9.655	10.344	11.909	12.279	13.843	14.785	16.346	17.673	18.100	
real	2.962	5.488	8.497	11.176	14.605	17.378	19.832	22.047	24.648	28.040	30.512	33.149	36.894	38.658	41.394	45.357	50.059	51.805	54.526	56.348
user	1.834	3.476	5.201	6.933	8.824	10.459	12.137	13.769	15.279	17.171	18.924	20.496	22.350	24.148	25.551	27.807	29.376	31.010	32.846	34.188
sys	0.930	1.720	2.796	3.612	4.580	5.505	6.525	7.001	8.002	8.953	9.840	10.691	11.683	12.344	13.527	14.805	15.882	16.958	18.098	18.598
real	2.887	5.562	8.569	11.474	14.729	17.739	20.018	21.760	25.010	28.163	30.528	33.127	36.338	39.284	41.896	44.976	49.899	50.735	54.082	56.906
user	1.729	3.419	5.224	7.084	8.871	10.485	12.184	13.816	15.354	17.094	18.920	20.495	22.350	23.889	25.563	27.805	29.395	30.943	32.750	34.485
sys	0.944	1.750	2.843	3.753	4.717	5.745	6.671	7.602	8.234	9.264	9.719	10.803	11.895	12.505	13.708	14.664	15.509	16.698	17.904	18.399
real	2.843	5.400	8.553	11.467	14.687	17.214	20.140	22.664	25.399	27.466	30.912	32.321	36.930	38.620	41.498	45.606	49.031	51.240	53.788	57.503
user	1.759	3.414	5.271	7.041	8.786	10.501	12.183	13.870	15.515	17.198	19.017	20.567	22.451	24.137	25.730	27.467	29.360	31.093	32.510	34.361
sys	0.924	1.707	2.754	3.735	4.706	5.530	6.733	7.422	8.342	9.029	10.051	10.814	11.817	12.244	13.380	14.677	15.363	17.121	17.274	18.829
real	2.781	5.742	8.541	11.545	14.681	17.223	20.444	22.139	24.844	27.852	30.339	34.288	35.584	37.840	42.274	45.203	48.929	51.129	53.445	57.642
user	1.709	3.510	5.261	6.988	8.880	10.374	12.089	13.701	15.354	17.229	18.889	20.850	22.259	23.824	25.571	27.855	29.346	30.883	32.719	34.804
sys	0.921	1.889	2.752	3.872	4.784	5.479	6.276	7.175	8.056	9.037	9.759	11.416	11.340	12.004	13.605	14.703	15.370	16.566	17.632	19.281
real	2.750	6.126	8.351	11.355	14.230	17.194	19.450	22.882	25.181	27.532	29.803	33.233	35.435	39.553	42.051	45.679	48.817	50.405	54.714	55.969
user	1.729	3.615	5.231	6.917	8.721	10.432	12.076	13.919	15.527	17.171	18.956	20.414	22.211	24.084	25.578	27.503	29.198	30.862	33.036	34.426
sys	0.874	2.083	2.655	3.774	4.369	5.360	6.267	7.594	8.196	8.824	9.213	10.937	11.275	13.161	13.549	14.742	15.399	16.491	18.348	18.800
real	2.915	5.442	8.503	11.528	14.590	17.538	19.797	22.139	25.452	27.772	28.926	35.095	39.880	41.720	45.115	48.139	50.953	53.425	56.366	
user	1.711	3.409	5.368	7.015	8.821	10.422	12.150	13.706	15.390	17.110	18.920	20.463	22.171	23.850	25.572	27.805	29.391	31.181	32.625	34.655
sys	0.983	1.753	2.793	3.768	4.657	5.716	6.642	7.169	8.195	8.932	9.985	10.950	11.408	12.200	13.448	15.116	17.077	19.485	20.513	
real	2.771	5.676	8.428	11.701	14.742	17.716	19.643	22.623	25.042	28.097	30.109	33.144	35.924	38.431	41.971	46.874	50.527	50.554	53.148	55.082
user	1.764	3.543	5.130	7.055	8.747	10.412	11.991	13.792	15.514	17.397	18.884	20.582	22.103	24.051	25.584	28.268	30.918	32.422	35.010	
sys	0.859	1.796	2.861	3.872	4.789	5.584	6.518	7.497	8.110	9.056	9.934	10.701	11.335	12.191	13.772	15.877	16.738	16.895	17.610	19.399
real average	2.8333	5.6825	8.5882	11.4650	14.6381	17.4778	19.9468	22.2521	24.9708	28.0134	30.2801	33.1991	36.288	38.8316	41.9889	45.4618	49.497	51.1557	54.0675	56.7815
user + sys average	2.6719	5.3461	8.0685	10.7481	13.4362	16.016	18.7493	20.9751	23.5434	26.1075	28.5771	31.3079	33.9438	36.5151	39.2808	42.5256	45.3511	47.8948	50.6339	53.0979
real average / dots	2.8333	2.84125	2.86273333	2.8664	2.92722	2.912966667	2.849557143	2.78151525	2.774511111	2.80134	2.752736364	2.766591667	2.791384615	2.773685714	2.79926	2.8413625	2.911582235	2.84198333	2.84567895	2.839075
user + sys average / dots	2.6719	2.67305	2.68895	2.687025	2.68724	2.689533333	2.678417429	2.6218875	2.615933333	2.61057	2.589718182	2.608991667	2.611061538	2.608221429	2.61872	2.65785	2.667711765	2.660282222	2.664942105	2.654895

эксперимент 2

2a.sh – для данного n параллельно считает значение функции в n точках (сам скрипт дожидается завершения последнего процесса):

```
#!/bin/bash
> prcs
((n=$1+1))
period=$(echo "scale=6; 2 / $n" | bc)
chmod +x lnn.sh
for ((i=1;i<$n;i++)); do
dot=$(echo "scale=6; -1 + $i * $period" | bc)
./lnn.sh $dot &
done
ps aux | grep -i "./lnn.sh" >> prcs
while read p; do
pid=$(echo $p | awk '{print $2}')
tail --pid=$pid -f /dev/null
done <> prcs
```

2b.sh – для каждого n от 1 до 15 10 раз запускает предыдущий скрипт и выводит время выполнения:

```
#!/bin/bash
> report.2b
chmod +x 2a.sh
for ((i=1;i<16;i++)); do
echo $i >> report.2b
for ((j=1;j<11;j++)); do
{ time ./2a.sh $i ; } 2>> report.2b
done
done
```

среднее арифметическое времени вычисления функции параллельно в n точках в зависимости от n

n	real average	user + sys average
1	3.5	3.5
2	3.5	4.5
3	3.5	8.0
4	3.5	8.5
5	3.5	8.5
6	3.5	8.5
7	3.5	8.5
8	3.5	8.5
9	3.5	8.5
10	3.5	8.5
11	3.5	8.5
12	3.5	8.5
13	3.5	8.5
14	3.5	9.0
15	3.5	9.5

процессоре процессор так или иначе параллелил вычисления, поэтому при данном подсчете средних значений `user+sys` превышает время `real`, которое, в свою очередь, все так же близко к изначальным 3 секундам на одну точку. параллельные вычисления на 2x процессорах работали в среднем на треть быстрее (очевидно, здесь есть что параллелизм)

эксперименты с параллельным и последовательным выполнением задач с большими объемами считываемых и сохраняемых данных

эксперимент 1

`arr.sh` – скрипт, последовательно умножающий и дописывающий в конец файла значения из него:

```
1 #!/bin/bash
2 c=0
3 while read line; do
4   ((c=c+1))
5   if [ $c -ge 15000 ]; then break fi
6   ((x=line*2))
7   echo $x >> $1
8 done <$1
```

`2.1b.sh` – для данного n последовательно вызывает `arr.sh` n раз:

```
1 #!/bin/bash
2 n=$1 k=$2
3 chmod +x arr.sh
4 for ((i=1;i<=$n;i++)); do
5   ./arr.sh f$i.$k
6 done
```

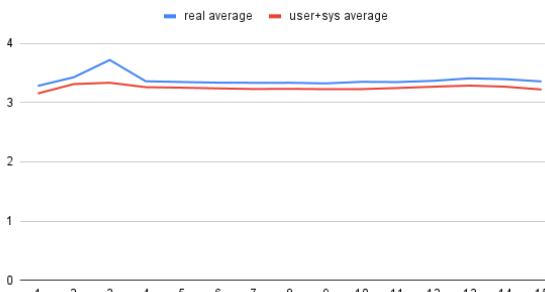
`2.1c.sh` – запускает `2.1b.sh` для каждого n от 1 до 15 10 раз и выводит время, затраченное на исполнение:

```
1 #!/bin/bash
2 > report.1c
3 chmod +x 1b.sh
4 chmod +x gen.sh
5 for ((i=1;i<16;i++)); do
6   echo $i >> report.1c
7   for ((j=1;j<11;j++)); do
8     { time ./1b.sh $i $j ; } 2>> report.1c
9   done
10  ./gen.sh
11 done
```

(`gen.sh` – просто генератор файлов):

```
1 #!/bin/bash
2 for ((i=1;i<16;i++)); do
3   for ((k=1;k<11;k++)); do
4     > f$i.$k
5   cp f1 f$i.$k
6   done
7 done
```

среднее время последовательной обработки файлов в зависимости от n



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
real	3.235	6.669	10.176	13.677	15.796	20.135	22.963	26.6	29.84	34.456	36.85	40.455	44.87	47.692	50.83
user+sys	3.13	6.417	9.753	13.243	15.297	19.598	22.32	25.8	28.987	32.954	35.777	39.178	43.526	46.136	48.297
real	3.247	6.725	10.385	13.591	15.644	20.511	22.654	26.235	30.298	34.356	36.632	40.232	43.681	46.91	51.085
user+sys	3.111	6.455	10.026	13.154	15.157	18.843	21.993	25.438	29.386	32.775	35.602	39.077	42.411	45.559	48.561
real	3.303	7.205	10.928	14.097	17.374	18.874	22.947	26.034	29.698	33.165	36.851	40.505	44.82	48.117	50.424
user+sys	3.216	6.946	9.758	13.644	16.9	19.303	22.267	25.265	28.848	32.067	35.797	39.282	43.563	46.671	47.547
real	3.161	7.161	10.216	14.082	18.782	19.77	23.355	26.153	30.716	33.15	37.062	40.86	44.423	47.995	50.047
user+sys	3.079	6.88	9.83	13.67	18.312	19.2	22.494	25.412	29.865	32.169	36.01	39.651	42.41	45.907	47.701
real	3.232	6.749	10.123	13.009	15.496	19.635	23.501	26.805	29.788	32.543	36.38	40.544	45.233	45.685	51.404
user+sys	3.103	6.532	9.733	12.631	15.006	19.021	22.302	26.047	28.975	31.489	35.33	39.358	43.164	44.407	49.02
real	3.24	6.82	10.694	13.587	15.681	20.104	23.435	26.839	29.495	33.256	36.02	40.754	43.175	47.172	49.628
user+sys	3.091	6.613	10.103	13.19	15.182	19.54	22.746	26.078	28.719	32.136	35.041	39.555	41.968	45.09	47.515
real	3.382	6.974	10.745	13.32	15.345	20.098	23.081	26.27	30.97	34.15	36.765	40.438	43.883	46.916	49.771
user+sys	3.153	6.754	10.32	12.938	15.895	19.532	22.395	25.496	29.865	32.978	35.711	39.201	41.925	45.535	48.318
real	3.306	6.544	13.865	13.159	16.35	19.911	24.387	27.349	29.503	32.658	37.309	40.469	44.883	47.548	51.632
user+sys	3.215	6.36	8.8	12.793	15.906	19.297	23.673	26.102	28.466	31.638	36.283	39.267	42.833	46.143	50.174
real	3.16	7.099	14.082	12.774	17.456	19.425	23.728	26.959	29.28	33.628	37.468	40.317	44.51	49.477	48.719
user+sys	3.078	6.873	10.682	12.428	15.943	18.888	23.034	26.214	28.514	31.901	35.802	39.126	43.137	46.443	47.385
real	3.562	6.632	10.405	13.131	15.503	20.764	23.524	27.609	29.662	33.779	36.878	39.637	43.84	48.017	50.145
user+sys	3.376	6.439	10.073	12.749	15.032	20.208	22.875	26.781	28.846	32.736	35.828	38.593	42.492	45.79	48.759
real average	3.2826	3.42885	3.720633333	3.360675	3.348654	3.337116687	3.3355625	3.324866667	3.351541	3.347409091	3.368425	3.410138462	3.396353714	3.3579	
user+sys average	3.1552	3.31345	3.335533333	3.261	3.2526	3.2405	3.229985714	3.2327875	3.227455556	3.22843	3.2471	3.269666667	3.287915385	3.26915	3.221846667

эксперимент 2

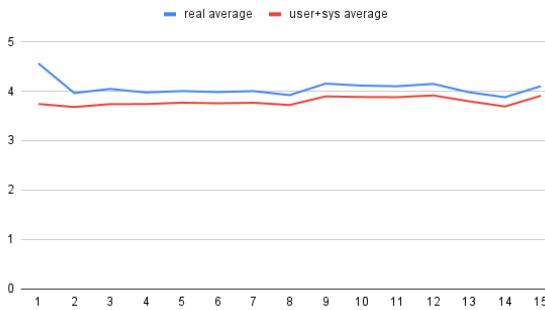
2.2a.sh – для данного n параллельно обрабатывает n файлов (сам скрипт дожидается завершения последнего процесса):

```
1 #!/bin/bash
2 > prcs
3 n=$1 k=$2
4 chmod +x arr.sh
5 for ((i=1;i<=$n;i++)); do
6 ./arr.sh f$i.$k &
7 done
8 ps aux | grep -i "./arr.sh" >> prcs
9 while read p; do
10 pid=$(echo $p | awk '{print $2}')
11 tail --pid=$pid -f /dev/null
12 done <prcs
```

2.2b.sh – для каждого n от 1 до 15 10 раз запускает предыдущий скрипт и выводит время выполнения:

```
1 #!/bin/bash
2 > report.2b
3 chmod +x 2a.sh
4 chmod +x gen.sh
5 for ((i=1;i<16;i++)); do
6 echo $i >> report.2b
7 for ((j=1;j<11;j++)); do
8 { time ./2a.sh $i $j ; } 2>> report.2b
9 done
10 ./gen.sh
11 done
```

среднее время обработки файла при n параллельных скриптах в зависимости от n

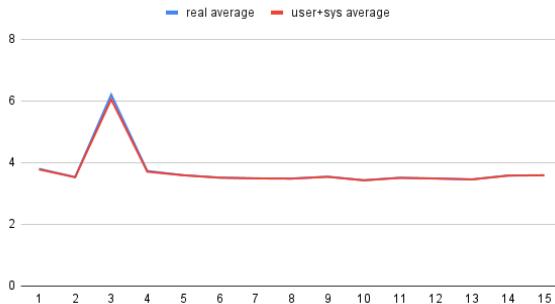


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
real	4.567	7.859	12.06	16.371	19.652	24.192	27.874	30.284	35.396	41.225	43.145	51.766	54.208	54.837	59.109
user+sys	3.559	7.504	10.946	14.263	16.399	22.984	26.278	29.164	33.994	37.196	40.905	47.772	50.947	52.24	56.496
real	4.575	7.809	12.258	16.302	19.563	24.883	28.122	31.944	37.519	41.409	46.707	51.039	60.175	53.237	62.626
user+sys	3.582	7.175	11.284	15.525	18.38	23.155	26.115	29.847	35.4	39.073	44.357	47.637	56.162	51.224	58.731
real	4.563	7.807	12.22	16.337	19.62	23.963	28.389	32.657	38.816	40.225	43.768	49.403	52.353	53.32	62.368
user+sys	3.603	7.318	11.471	14.917	18.669	23.027	27.373	30.562	36.599	38.112	41.425	47.095	50.044	50.722	59.567
real	4.552	7.821	12.065	16.312	19.676	23.778	28.508	31.602	38.19	39.947	43.529	49.021	48.252	53.006	62.855
user+sys	3.652	7.177	10.987	14.86	18.383	22.787	26.721	29.808	35.777	38.131	41.663	46.318	45.891	50.675	60.018
real	4.594	7.871	11.065	15.902	20.767	23.843	27.434	30.306	37.003	39.761	46.126	48.564	50.072	53.029	62.483
user+sys	3.701	7.322	10.684	15.278	19.255	22.514	25.992	29.054	34.678	38.043	43.471	46.182	48.276	50.713	59.791
real	4.56	7.797	12.193	16.399	20.828	23.459	28.29	31.287	37.02	39.923	45.759	49.244	49.707	52.725	62.51
user+sys	4.176	7.426	11.374	15.119	19.483	21.913	26.307	30.066	34.616	38.243	43.441	46.493	47.706	50.938	59.384
real	4.588	7.758	12.071	15.321	20.842	22.864	27.248	31.335	36.872	42.484	45.743	51.961	49.337	55.311	60.387
user+sys	3.588	7.481	10.895	14.736	19.269	21.486	25.755	29.869	34.551	39.403	42.834	48.913	47.733	51.842	57.511
real	4.504	8.916	12.087	15.359	19.874	23.916	27.265	31.296	37.001	41.43	45.271	50.356	48.648	55.672	60.707
user+sys	4.277	7.715	11.286	14.747	18.684	22.197	25.735	29.987	34.709	38.805	42.708	47.265	47.063	53.199	57.784
real	4.56	7.828	12.197	15.282	18.596	23.841	28.389	31.531	37.853	42.672	44.475	48.41	53.52	52.916	60.4
user+sys	3.646	7.113	11.345	14.636	17.93	22.077	26.346	29.683	35.763	40.107	42.124	46.357	50.6	50.274	57.776
real	4.564	7.772	13.156	15.342	20.673	24.199	28.549	31.351	38.089	42.284	46.438	48.009	50.833	58.758	61.796
user+sys	3.621	7.352	11.89	14.277	19.87	23.149	26.928	29.54	34.406	39.078	43.695	45.668	48.903	54.755	59.007
real average	4.5625	3.9619	4.045733333	3.973175	4.00222	3.9823	4.000971429	3.9199125	4.152877778	4.1136	4.099645455	4.148108333	3.977730769	3.877221429	4.101606667
user+sys average	3.7403	3.67915	3.737733333	3.73895	3.76684	3.754816667	3.765	3.71975	3.894366667	3.88191	3.878390909	3.914166667	3.794087692	3.689851743	3.9071

эксперимент 3

эксперимент 1 на двух процессорах:

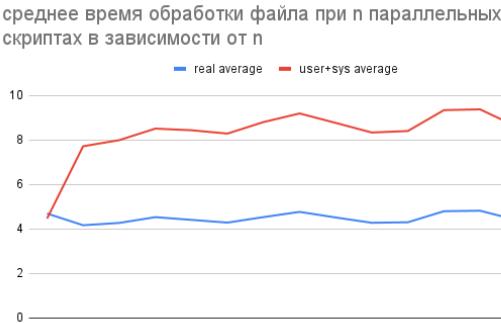
среднее время обработки файлов в зависимости от n



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
real	3.765	7.099	10.563	13.532	17.729	22.059	23.755	27.308	31.951	37.483	39.673	39.832	45.399	47.638	52.946
user+sys	3.748	7.09	10.554	13.522	17.696	22.049	23.716	27.284	31.934	37.445	39.636	39.795	45.364	47.594	52.878
real	4.918	6.829	10.16	14.025	18.505	20.616	24.612	27.549	31.49	34.273	40.399	41.704	45.516	49.626	54.047
user+sys	4.755	6.823	10.146	14.014	18.495	20.598	24.597	27.538	31.473	34.263	40.367	41.672	45.476	49.614	53.999
real	3.472	6.546	14.225	14.312	17.999	23.848	24.585	27.186	30.518	34.602	41.84	40.09	45.683	52.529	54.698
user+sys	3.469	6.543	14.014	14.279	17.987	23.708	24.576	27.146	30.501	34.591	41.786	40.052	45.636	52.489	54.651
real	3.696	7.027	22.808	13.661	18.67	20.407	24.873	28.355	32.946	33.845	38.442	40.111	44.898	49.654	58.415
user+sys	3.695	7.024	22.125	13.647	18.659	20.385	24.845	28.193	32.772	33.813	38.425	40.078	44.861	49.601	58.212
real	3.474	7.803	22.409	18.411	18.153	19.816	24.377	27.198	30.391	33.874	37.919	41.868	44.152	49.772	51.559
user+sys	3.472	7.79	21.688	18.224	18.14	19.812	24.362	27.189	30.374	33.862	37.892	41.817	44.119	49.738	51.523
real	3.713	6.798	23.096	17.818	17.83	20.567	24.969	27.438	31.658	34.104	38.029	43.679	45.187	49.998	58.517
user+sys	3.707	6.796	22.366	17.152	17.818	20.549	24.953	27.417	31.627	34.086	38.007	43.653	45.184	49.957	58.272
real	3.843	7.444	24.766	14.691	18.336	20.639	24.216	27.867	31.684	33.825	37.773	41.055	44.263	51.731	54.324
user+sys	3.839	7.44	23.934	14.679	18.328	20.623	24.21	27.856	31.671	33.808	37.749	41.03	44.239	51.701	54.253
real	3.883	7.273	23.304	14.638	17.93	21.657	24.346	27.91	33.295	34.688	37.626	40.892	46.294	52.222	51.974
user+sys	3.879	7.266	22.618	14.632	17.922	21.645	24.332	27.897	33.231	34.668	37.605	40.87	46.259	52.192	51.925
real	3.697	6.964	22.976	13.814	16.62	20	24.796	28.657	32.777	33.13	36.832	45.209	43.615	49.448	51.21
user+sys	3.694	6.944	22.245	13.808	16.609	19.982	24.774	28.636	32.763	33.11	36.807	45.146	43.587	49.417	51.183
real	3.539	6.833	11.859	14.304	17.87	21.321	23.832	29.275	32.238	33.06	37.43	43.08	44.476	48.918	51.308
user+sys	3.535	6.831	11.832	14.289	17.857	21.303	23.823	29.492	32.214	33.053	37.4	43.931	44.445	48.9	51.278
real average	3.8	3.5308	6.205333333	3.73015	3.59284	3.5155	3.490585714	3.4684125	3.543755556	3.42864	3.5080754545	3.48675	3.457961538	3.581685714	3.59332
user+sys average	3.7793	3.52735	6.050733333	3.705675	3.59022	3.510883333	3.4884	3.4806	3.539555556	3.42699	3.506127273	3.4837	3.455038462	3.5797235714	3.587826667

эксперимент 4

эксперимент 2 на двух процессорах:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
real	4.677	7.961	13.493	17.558	21.813	27.312	29.281	40.502	40.284	48.591	49.855	50.822	63.925	62.368	78.195
user+sys	4.282	15.535	24.568	33.62	41.463	52.908	57.372	77.847	77.803	95.248	95.648	99.671	125.658	121.092	152.801
real	4.544	8.954	12.535	17.859	20.965	25.077	28.93	35.796	40.955	42.429	47.956	54.413	68.144	61.254	76.091
user+sys	4.293	16.204	23.184	33.454	40.059	49.149	57.12	70.284	79.7	82.245	98.075	103.929	130.26	119.826	148.883
real	4.544	7.934	12.014	17.758	22.247	25.086	32.333	39.201	41.015	43.285	46.688	54.798	66.356	59.417	75.905
user+sys	4.177	14.68	22.226	31.927	41.375	48.553	62.011	75.945	78.943	83.757	92.245	106.86	128.698	116.387	149.37
real	4.545	7.91	12.075	18.12	22.045	25.029	30.516	40.98	41.386	41.102	46.296	54.727	63.36	58.846	69.134
user+sys	4.244	15.053	23.218	32.159	41.872	48.9	60.114	78.225	80.469	84.465	90.994	106.577	121.888	114.733	133.931
real	4.56	7.881	13.268	16.379	21.973	24.811	32.324	37.924	39.61	41.934	45.782	54.543	59.491	58.979	65.997
user+sys	4.277	15.074	24.312	31.988	42.15	48.364	63.568	74.404	77.743	82.265	90.633	107.256	117.312	116.556	128.352
real	5.509	8.914	12.233	18.413	22.327	26.113	32.634	36.788	38.981	41.963	47.424	63.351	60.818	61.312	63.648
user+sys	5.733	16.024	23.411	35.717	43.234	49.912	62.789	69.51	76.359	81.87	91.71	123.443	118.029	119.588	124.686
real	4.82	7.792	12.074	18.499	22.705	26.383	33.245	38.17	42.84	44.223	47.4	57.959	58.847	61.764	66.765
user+sys	4.624	14.77	24.357	35.649	43.614	51.096	64.396	73.362	82.304	81.397	91.969	113.816	115.695	119.865	130.386
real	4.594	9.002	13.157	18.456	20.811	25.002	33.628	36.876	39.985	41.583	46.166	61.132	64.585	60.819	66.395
user+sys	4.313	16.075	24.562	34.656	44.898	48.297	64.609	70.551	78.614	74.127	89.809	119.5	126.742	118.088	129.478
real	4.583	8.014	13.283	19.754	22.633	25.99	32.11	36.73	39.881	43.001	46.469	62.93	64.942	62.325	66.868
user+sys	4.499	15.104	25.093	36.97	42.576	49.481	61.755	71.164	77.414	83.213	90.506	120.445	122.463	131.219	
real	4.574	9.037	13.169	18.543	23.075	26.463	32.502	36.839	41.72	42.908	46.574	61.357	56.433	70.091	66.139
user+sys	4.249	16.075	24.562	34.656	44.898	50.637	62.705	74.127	79.822	80.91	91.947	119.435	109.21	137.188	130.648
real average	4.695	4.16995	4.27336667	4.5333	4.41188	4.28776667	4.535757143	4.770075	4.518111111	4.27999	4.30372773	4.80026667	4.822315385	4.408692557	4.634246667
user+sys average	4.4691	7.71685	7.993	8.5127	8.4382	8.28823333	8.806271429	9.192737	8.76866667	8.33562	8.405054545	9.34276667	9.3771	8.012757143	9.065026667

ВЫВОДЫ

lorem ipsum в экспериментах 1 и 3 отличий от экспериментов первой части мало, нагрузка на процессор все так же маленькая при последовательном запуске `arr.sh`, и параметр n в данных экспериментах ни на что особо не влияет (не смотрим на скачок при n равном 3, его я не могу объяснить никак кроме "случайных" процессов os). распределять работу по двум процессорам все так же нет особой надобности, поэтому значения времени работы в 3 эксперименте аналогичны значениям первого. в параллельных экспериментах на двух процессорах происходит ровно то же самое, что в экспериментах первой группы, но на одном от количества параллельных процессов не менялось ничего, также, как если бы все скрипты запускались последовательно (еще и на секунду дольше). (?) получается, на одном процессоре никакой многопоточности в работе с большими файлами, в отличие от вычислительно сложных задач, так просто устроить нельзя