**Міністерство освіти і науки України**

**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з контрольної роботи №2

з дисципліни «Моделювання та Аналіз Програмного Забезпечення»

**Виконав:**

Студент групи КНТ-122 О. А. Онищенко

**Прийняли:**

Викладач: Ж. К. Камінська

2024

Опис завдання

Скласти модель роботи верстата за таких умов:

1. Надходження деталей на верстат відбувається за законом Пуассона із середньою інтенсивністю 2 деталі на годину

2. 55% деталей є деталями типу 1, 30% - типу 2 і решта 15% - типу 3

3. Час обробки деталей типів 1, 2 і 3 розподілено експоненційно із середніми значеннями в 20, 30 і 40 хв відповідно

4. Для визначення дисципліни обслуговування на верстаті використовується таке правило: на початку обслуговуються деталі, які прочекали 90 хв і більше; серед цих деталей використовується критерій динамічного пріоритету

Бо так полюбив Бог світ, що Він віддав Сина Свого Однородженого (Ісуса), аби кожен хто вірує в Нього, не згинув, але мав вічне життя. ([Йоан 3:16](https://www.bible.com/uk/bible/compare/JHN.3.16))

Код програми

#include "../simc/simc.h"

#include <iostream>

using namespace std;

void Set\_Dynamic\_Priority(ptransact Current\_Transact) {

  const int Max\_Priority=7;

  const int Min\_Priority=1;

  Current\_Transact->prty=randab(Min\_Priority, Max\_Priority, v1);

}

void Insert\_Custom\_List(plistt Custom\_List) {

  outtlist(current);

  inlt(Custom\_List, trans);

  if (trans->pi[2]>=90) Set\_Dynamic\_Priority(trans);

  trans->testprty = false;

  if (trans->pi[2]>=90) Custom\_List->first = Custom\_List->first->sled;

  else Custom\_List->first = Custom\_List->first->sled->sled;

  trans = NULL;

}

void solve() {

  auto Modeling\_Hours=7;

  auto Modeling\_Time=Modeling\_Hours\*60;

  pfacility Machine;

  plistt List;

  initlist(Modeling\_Time);

  initcreate(1, 0);

  newfac(Machine, "\"Machine\"");

  newuserlt(List, "\"List\"");

  float First\_Delay=20.0;

  float Second\_Delay=30.0;

  float Third\_Delay=40.0;

  auto Average\_Interval=2\*60.0;

  while (systime < Modeling\_Time) {

    plan();

    switch (sysevent) {

      case 1: create(randpoisson(Average\_Interval, v7)); trans->pi[2]+=1; break;

      case 2: if (rand01(v1) >= 0.55) { trans->pi[1] = 1; } else if (rand01(v1) >= 0.30) { trans->pi[1] = 2; } else { trans->pi[1] = 3; } break;

      case 4: Insert\_Custom\_List(List); outuserlt(List); break;

      case 5: seize(Machine); break;

      case 6: if (trans->pi[1]==1) { delayt(randexp(First\_Delay,v1)); } else if (trans->pi[1]==2) { delayt(randexp(Second\_Delay,v1)); } else { delayt(randexp(Third\_Delay,v1)); } break;

      case 8: outfac(Machine); break;

      case 9: destroy(); break;

    }

  }

  cout << "Modeling finished, praise Jesus Christ our Holy Lord GOD Almighty King of Kings and Lord of Lords" << endl << endl;

  printall();

}

int main() {

  solve();

  return 0;

}

Виконання програми

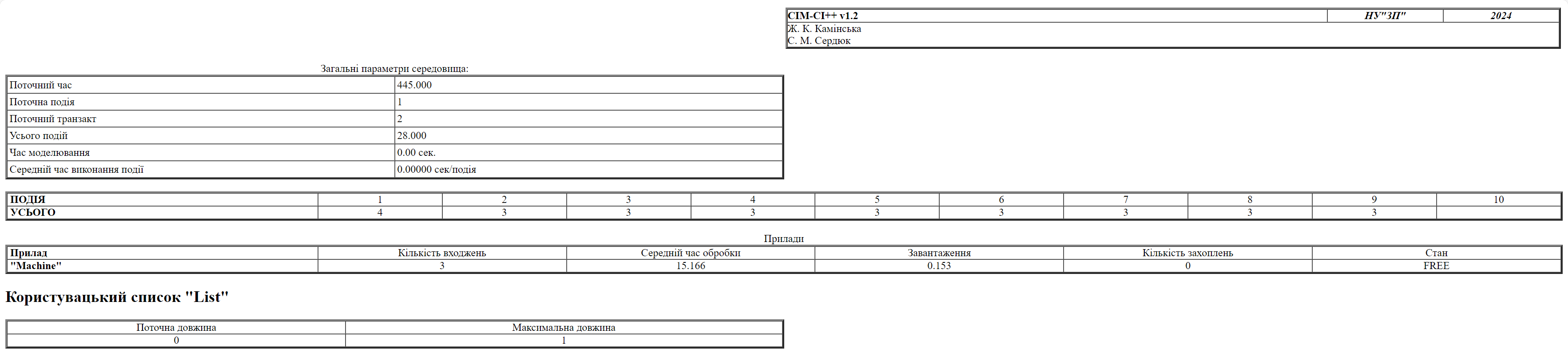


Рисунок 1.1 – Браузер



Рисунок 1.2 – Консолька ♰