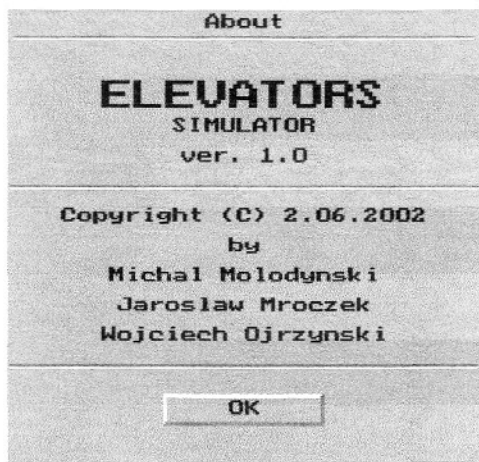


# Elevators Simulator ver 1.0

## Instrukcja



# 1. Wprowadzenie

Pakiet **Elevators Simulator** stworzony został w celu usprawnienia prac symulacyjnych, dotyczących pracy wind w wielopiętrowych budynkach. Dzięki swej elastyczności, program znajduje zastosowanie zarówno przypadku małych budynków o niewielkiej ilości wind, jak i w wieżowcach o bardzo zróżnicowanym natężeniu ruchu między piętrami. Głównym zadaniem programu jest pomoc, w wyznaczeniu optymalnego algorytmu obsługi oraz optymalnej ilości wind dla danego systemu opisanego przez takie parametry jak: ilość pięter, częstotliwość odwiedzania piętra, częstotliwość opuszczania piętra, średni czas przejazdu windy pomiędzy dwoma sąsiednimi piętrami, średni czas postoju na piętrze niepustej windy, maksymalna liczba pasażerów windy, maksymalna liczba oczekujących. Program wyposażony jest w interfejs graficzny umożliwiający obserwację pracy systemu oraz kontrolę zachowania się systemu w sytuacjach ekstremalnych. Możliwe jest również przeprowadzenie szybkiej symulacji w tle, której wynikiem jest zestaw wykresów: średniego i maksymalnego czasu oczekiwania oraz łącznej ilości przejazdów wind pomiędzy piętrami w zależności od liczby wind, będących wyznacznikiem niezawodności i sprawności systemu.

W skład pakietu wchodzi:

- program instalacyjny
- program symulacyjny
- moduł algorytmów
- kompilator

Do poprawnego funkcjonowania pakietu niezbędne są następujące pliki i foldery:

**ELVSYM.EXE** - program symulacyjny

**EGAVGA.BGI** - standardowy sterownik grafiki zapożyczony z pakietu Borland Pascal

**INSTALL.EXE** - program instalacyjny

**ALG** - folder roboczy modułu algorytmów, w skład którego wchodzi pliki:

**ALGS.PAS** - moduł algorytmów

**ELVCPL.EXE** - kompilator modułu algorytmów

**TPC.EXE** - standardowy kompilator zapożyczony z pakietu Borland Pascal

**TURBO.TPL** - standardowa biblioteka kompilatora TPC.EXE

**EGAVGA.BGI** - standardowy sterownik grafiki z pakietu Borland Pascal

**GRAPH.TPU** - standardowy moduł graficzny z pakietu Borland Pascal

**ELVSYM.DAT** - plik danych dla kompilatora ELVCMP.EXE

**WINDOWS.TPU** - moduł procedur graficznych

**BUILD.TPU** - moduł odpowiedzialny za budynek

**FLIST.TPU** - moduł odpowiedzialny za listę zgłoszeń

**SYM.TPU** - moduł odpowiedzialny za symulację

**MOUSE.TPU** - moduł odpowiedzialny za obsługę myszy

**VAR.S.TPU** - moduł odpowiedzialny za zmienne globalne

**INSTALL** - folder niezbędny do poprawnego działania programu instalacyjnego, zawiera:

**COMMAND.COM** - interpreter poleceń

**BLD** - folder danych o budynkach, zdefiniowanych przez użytkownika

**FLS** - folder danych o listach zgłoszeń, zdefiniowanych przez użytkownika

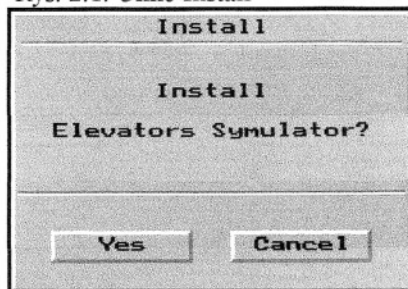
**RST** - folder wyników symulacji

## 2. Program instalacyjny

Program instalacyjny **install.exe** znajduje się w katalogu głównym pakietu **ElvSym**. Uruchomienie go powoduje skopiowanie „czystego” pakietu (bez plików roboczych) pod wskazaną ścieżkę. Do poprawnego jego działania niezbędny jest interpreter poleceń **command.com** zawarty w folderze **Install**,

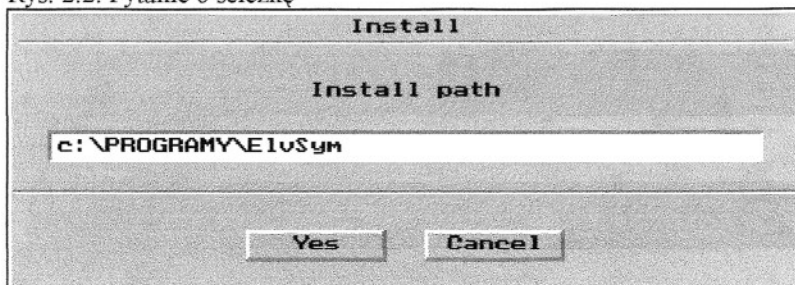
Bezpośrednio po uruchomieniu **install.exe** na ekranie pojawia się okno:

Rys. 2.1. Okno Install



Po kliknięciu na klawisz **Yes** program pyta o ścieżkę instalacji:

Rys. 2.2. Pytanie o ścieżkę



Wprowadzenie ścieżki następuje po kliknięciu na obszar edycyjny okna. Podana ścieżka musi istnieć fizycznie na dysku. Nie dotyczy to ostatniego folderu, który tworzony jest automatycznie, jeśli nie zostanie odnaleziony. Po kliknięciu na klawisz **Yes** następuje kopiowanie plików. Ze względu na niewielki rozmiar pakietu (1MB) trwa to nie więcej niż sekundę. W przypadku wprowadzenia niewłaściwej ścieżki zostanie wyświetlony odpowiedni komunikat o błędzie.

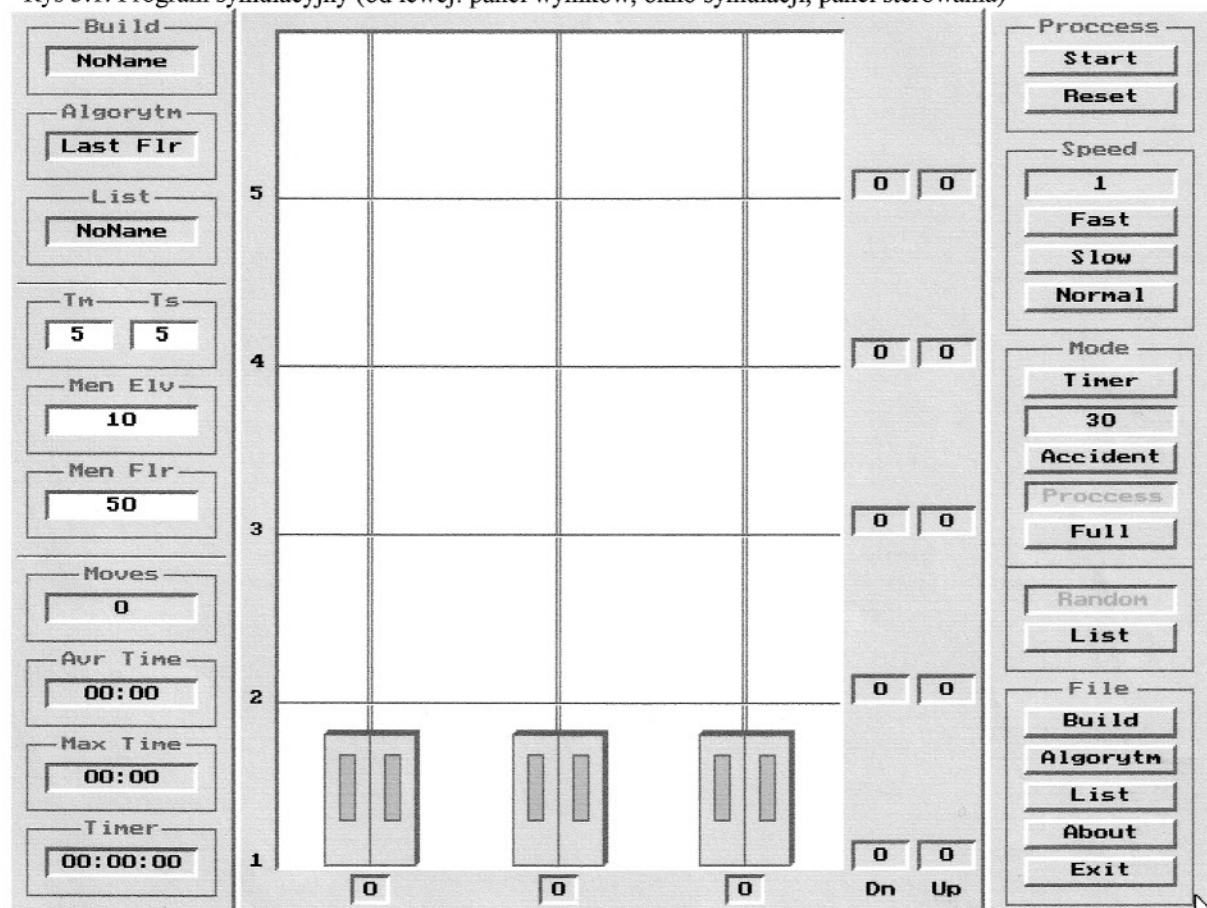
Kliknięcie na klawisz **Cancel** w każdym oknie powoduje natychmiastowe zakończenie działania programu.

### 3. Program symulacyjny

Program symulacyjny **elvsym.exe** znajduje się w folderze głównym pakietu **ElvSym**. Jego zadaniem jest przeprowadzanie symulacji oraz umożliwianie porównywania wyników symulacji. Po uruchomieniu programu na ekranie pojawia się wizytówka pakietu oraz interfejs użytkownika, w którego skład wchodzi:

- panel wyników
- okno symulacji
- panel sterowania

Rys 3.1. Program symulacyjny (od lewej: panel wyników, okno symulacji, panel sterowania)



#### 3.1. Panel wyników

W skład panelu wyników wchodzi (od góry): pole nazw plików, pole parametrów symulacji, pole wyników symulacji.

Pole nazw plików zawiera pola tekstowe **Build**, **Algorytm** i **List**. W polu wyświetlana jest nazwa pliku, zawierającego informacje o aktualnie badanym budynku. Jeśli nie został wczytany żaden plik, badanym budynkiem jest budynek domyślny **NoName**. Pole **Algorytm** zawiera nazwę aktywnego algorytmu obsługi wind. Domyślnie jest nim pierwszy algorytm z listy algorytmów. Pole **List** informuje o nazwie aktualnie używanego zbioru zgłoszeń. W przypadku nie wybrania żadnego zbioru jest nim zbiór **NoName** wygenerowany przez użytkownika.

Pole parametrów symulacji składa się z czterech pól edycyjnych. Zmianę parametrów symulacji realizuje się poprzez kliknięcie na dowolne z nich. Jeśli symulacja nie jest włączona możliwa jest edycja następujących parametrów:

**Tm** – średni czas przejazdu windy pomiędzy dwoma sąsiednimi piętrami

**Ts** – średni czas postoju niepustej windy na piętrze

**Men Elv** – maksymalna liczba pasażerów w windzie

**Men Flr** – maksymalna liczba oczekujących na windę na jednym piętrze

Pole wyników symulacji zawiera następujące pola informacyjne:

**Moves** – łączna liczba przejazdów wszystkich wind między sąsiednimi piętrami zliczana przez cały czas trwania symulacji

**Avr Time** – średni czas oczekiwania na windę (min:sec)

**Max Time** – maksymalny czas oczekiwania na windę (min:sec)

**Timer** – licznik czasu symulacji (godz:min:sec)

### 3.2. Okno symulacji

Okno symulacji jest graficznym interfejsem umożliwiającym obserwację pracy wind i badanie zachowanie się systemu w różnych sytuacjach. Okno zawiera rysunek schematyczny budynku, przedstawiający pozycję i status wszystkich wind. Winda jest otwarta lub częściowo otwarta oznacza, że jest ona niegotowa do jazdy (czas postoju windy nie przekroczył **Ts**). Pola informacyjne pod windami zawierają aktualną liczbę pasażerów każdej windy. Wyświetlenie tej wartości na czerwono oznacza brak wolnych miejsc w windzie. Pola **Dn** i **Up** zawierają liczbę zgłoszeń na danym piętrze odpowiednio w dół i w górę. Jeśli symulacja nie jest włączona to możliwe jest ustawianie wind na pozycjach początkowych poprzez klikanie na pozycjach nad ustawianymi windami.

### 3.3. Panel sterowania

Panel sterowania umożliwia użytkownikowi sterowanie symulacją oraz zapis i odczyt informacji o budynkach i listach zgłoszeń. Składa się on z następujących klawiszy i pól edycyjnych:

**Start/Stop** – Włączanie i przerywanie symulacji. Kliknięcie na przycisk **Stop** nie powoduje zatrzymania symulacji tylko jej przerwanie (pauzowanie). Przycisk **Start** wyłącza tryb edycji parametrów symulacji i włącza tryb symulacji. Jeśli jest wybrany tryb **Full** przycisk **Start** uruchamia symulację pełną.

**Reset** – Zerowanie symulacji. Wyzerowanie wszystkich liczników i przywrócenie wartości początkowych symulacji. Jeśli symulacja jest zatrzymana **Reset** powoduje jej wyłączenie i przejście do trybu edycji parametrów.

**Speed** – Pole prędkości symulacji. W skład jego wchodzi pole informacyjne oraz klawisze **Fast**, **Slow**, **Normal**. Pole informacyjne zawiera aktualnie wybraną prędkość symulacji gdzie **1** oznacza symulację w czasie rzeczywistym.

**Fast** – Dwukrotne przyspieszenie symulacji. Maksymalna prędkość symulacji wynosi 16x czas rzeczywisty.

**Slow** – Dwukrotne zwolnienie symulacji. Minimalna prędkość symulacji wynosi czas rzeczywisty/16.

**Normal** – Ustawienie prędkości symulacji na zgodną z czasem rzeczywistym.

**Timer** – Ustawienie trybu symulacji **Timer**. W trybie tym symulacja zatrzymywana (pauzowana) jest co stały okres czasu określony w oknie edycyjnym poniżej. Okno to jest aktywne tylko w przypadku wybrania trybu **Timer**. Okres pracy do zatrzymania podawany jest w sekundach.

**Accident** – Ustawienie trybu symulacji **Accident**. W trybie tym symulacja zatrzymywana jest w przypadku wystąpienia zdarzenia takiego jak: zatrzymanie się windy na piętrze, wezwanie windy.

**Process** – Ustawianie ciągłego trybu pracy wind.

**Full** – Wybranie trybu symulacji pełnej (błyskawicznej).

**Random** – Wybranie trybu losowego występowania zgłoszeń (obserwacja pracy wind).

**List** – Wybranie trybu występowania zgłoszeń na podstawie listy zgłoszeń (badanie zachowanie systemu w sytuacjach ekstremalnych).

**Build** – Uaktywnienie okna **Build** umożliwiającego edycję oraz zapis i odczyt parametrów badanego budynku.

**Algorytm** – Uaktywnienie okna **Algorytm** i wybór algorytmu obsługi wind z listy dostępnych algorytmów.

**List** – Uaktywnienie okna **List** umożliwiającego edycję oraz zapis i odczyt listy zgłoszeń.

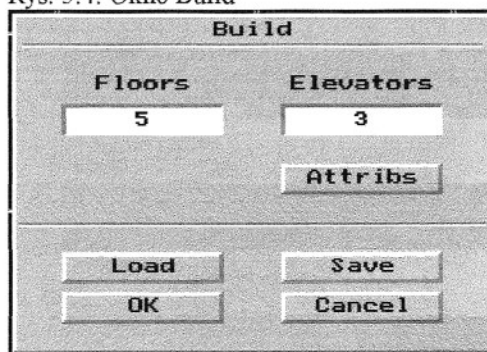
**About** – Wyświetlenie wizytówki programu **Elevator Simulator**.

**Exit** – Zakończenie działania programu.

### 3.4. Okno Build

Okno **Build** uaktywniane jest poprzez kliknięcie na klawisz **Build** w panelu sterowania. Jego zadaniem jest umożliwienie edycji oraz zapisu i odczytu parametrów budynku.

Rys. 3.4. Okno Build



Okno **Build** zawiera następujące klawisze i okna edycyjne:

**Floors** – Okno edycyjne zawierające liczbę pięter badanego budynku

**Elevators** – Okno edycyjne zawierające liczbę wind w badanym budynku. Wartość ta jest nieistotna w trybie symulacji pełnej.

**Attribs** – Uaktywnienie okna edycji atrybutów pięter.

**Load** – Wczytanie z pliku parametrów budynku.

**Save** – Zapis do pliku parametrów budynku.

**OK.** – Akceptacja wprowadzonych zmian.

**Cancel** – Rezygnacja



Do parametrów budynku zapisywanych i odczytywanych z plików należą: liczba pięter **Floors**, liczba wind **Elevators**, atrybuty pięter, czas przejazdu i postoju windy **Tm** i **Ts**, maksymalna liczba pasażerów **Men Elv**, maksymalna liczba oczekujących **Men Flr**, wartości początkowe pozycji wind.

### 3.5. Okno Attributes

Okno **Attributes** uaktywniane jest poprzez kliknięcie na klawisz **Attribs** w oknie **Build**. Okno to umożliwia edycję atrybutów badanego budynku.

Rys. 3.5. Okno Attributes

	From	To		From	To
1	1.000	1.000	4	1.000	1.000
2	1.000	1.000	5	1.000	1.000
3	1.000	1.000			

OK Cancel

Liczba par atrybutów **From** i **To** w oknie jest równa liczbie pięter wprowadzonej w polu edycyjnym **Floors** w oknie **Build**. Atrybut **From** jest równy średniej liczbie zgłoszeń na danym piętrze w czasie jednej minuty. Atrybut **To** wyraża częstotliwość odwiedzania piętra. Prawdopodobieństwo zdarzenia, że zgłoszenie będzie na i-te piętro wynosi  $To[i]/Suma\_To[i]$ , gdzie  $To[i]$  jest atrybutem **To** dla i-tego piętra a  $Suma\_To$  jest sumą wszystkich pozostałych atrybutów **To**. Klawiszem **OK**. akceptujemy wprowadzone zmiany, klawiszem **Cancel** przywracamy poprzednie wartości.

### 3.6. Okno Algorytm

Okno **Algorytm** uaktywniane jest poprzez kliknięcie na klawisz **Algorytm** w panelu sterowania. Umożliwia ono wybór algorytmu obsługi wind z listy dostępnych algorytmów.

Rys. 3.6. Okno Algorytm

Algorytm

Last Flr  
Next Flr  
Down-Up  
Empty

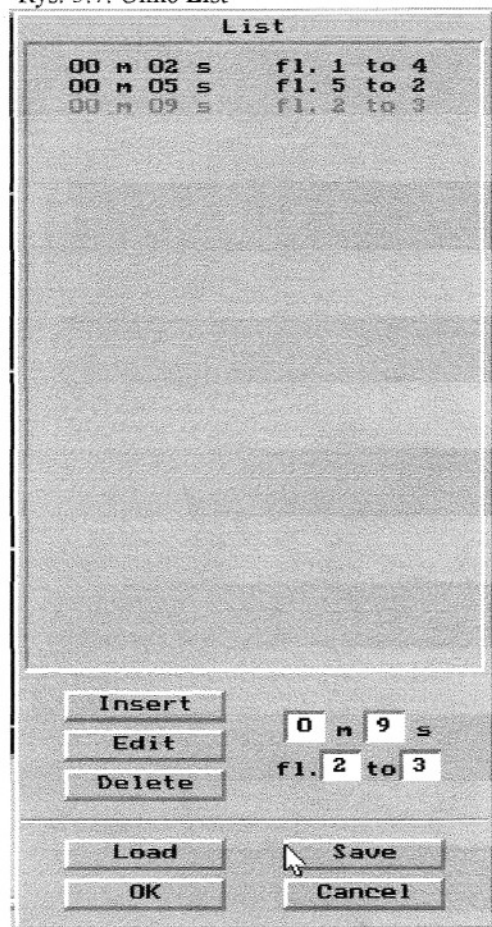
OK Cancel

Wybór algorytmu następuje poprzez kliknięcie na nazwę identyfikującą dany algorytm. Lista algorytmów wyświetlona w tym oknie jest listą nazw zdefiniowanych w module algorytmów. Nazwa wybranego algorytmu podana jest w oknie informacyjnym pod listą.

### 3.7. Okno List

Okno **List** uaktywniane jest poprzez kliknięcie na klawisz **List** w polu **File** panelu sterowania. Umożliwia ono edycję oraz zapis i odczyt listy zgłoszeń dla trybu symulacji **List**.

Rys. 3.7. Okno List



Wprowadzanie zgłoszeń realizuje się poprzez edycję danych w polach edycyjnych i kliknięcie na klawisz **Insert**. Czas wystąpienia zgłoszenia określamy w polach **m.** i **s** w formacie gdzie pierwszą wartością są minuty a drugą sekundy czasu wystąpienia zgłoszenia. Piętro, na którym występuje zgłoszenie określa pole **f1.** a piętro docelowe pole **to**. Edycję wprowadzonego zgłoszenia realizuje się poprzez wybranie za pomocą myszy danego zgłoszenia, edycję wartości w polach edycyjnych i kliknięcie na klawisz **Edit**. Klawisz **Delete** usuwa wybrane zgłoszenie.

Listę zgłoszeń można wczytać z pliku klikając na klawisz **Load** lub zapisać wygenerowaną listę za pomocą opcji **Save**. **OK.** akceptuje wprowadzone zmiany, **Cancel** przywraca listę zgłoszeń z przed edycji.



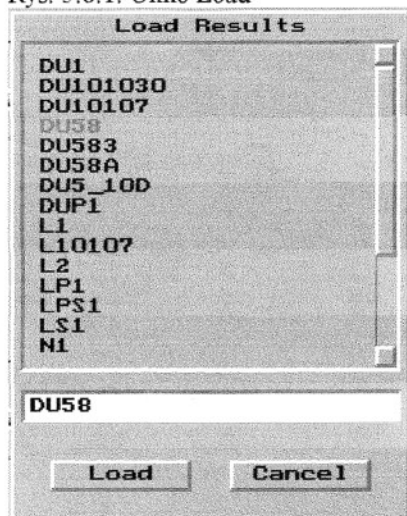
### 3.8. Odczyt/zapis do pliku

Program **Elevators Simulator** umożliwia odczyt/zapis do pliku:

- parametrów budynku w oknie **Build**
- listy zgłoszeń w oknie **List**
- wyników pełnej symulacji

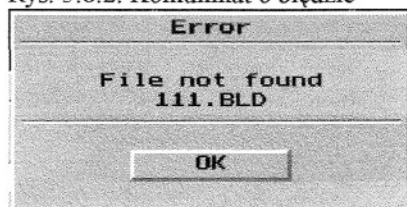
W każdym z powyższych przypadków wykorzystany jest ten sam komponent.

Rys. 3.8.1. Okno Load



Wybranie pliku do odczytu/zapisu realizowane jest przez kliknięcie na nazwę istniejącego pliku lub wpisanie nazwy pliku w polu edycyjnym pod listą plików. W przypadku próby odczytu z nieistniejącego pliku zostanie wyświetlony komunikat o błędzie.

Rys. 3.8.2. Komunikat o błędzie

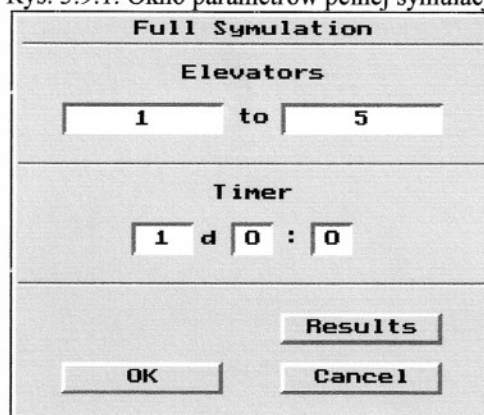


Próba odczytu pliku bez wybrania ani wprowadzenia nazwy pliku jest równoznaczna z rezygnacją podobnie jak w przypadku kliknięcie na klawisz **Cancel**.

### 3.9. Symulacja pełna (błyskawiczna)

Symulacja pełna umożliwia błyskawiczne przeprowadzenie badań porównawczych dla zadanego systemu i różnej ilości wind. Uruchamia się ją poprzez wybranie trybu **Full** i kliknięcie na przycisk **Start**. Symulacja pełna przeprowadzana w trybie **Random** niezależnie od liczby wind wprowadzonej w polu **Elevators** okna **Build**. Przedział liczby wind i czas symulacji dla każdego z systemów ustawiany jest przed rozpoczęciem symulacji w oknie startowym.

Rys. 3.9.1. Okno parametrów pełnej symulacji



Full Simulation

Elevators

1 to 5

Timer

1 d 0 : 0

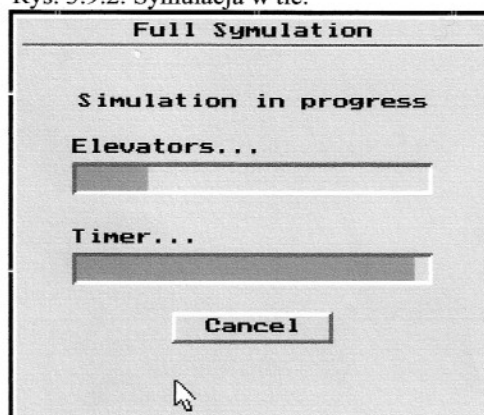
Results

OK Cancel

Zawartość okien edycyjnych **Elevators** określa minimalną i maksymalną liczbę wind w badanym systemie. Okna **Timer** zawierają czas pojedynczej symulacji w formacie dni.godz:min.

Klawisz **Results** służy do pominięcia symulacji w celu obejrzenia zapisanych wyników. Klawisz **Cancel** zamyka okno. Klawisz **OK** uruchamia symulację w tle.

Rys. 3.9.2. Symulacja w tle.



Full Simulation

Simulation in progress

Elevators...

Timer...

Cancel

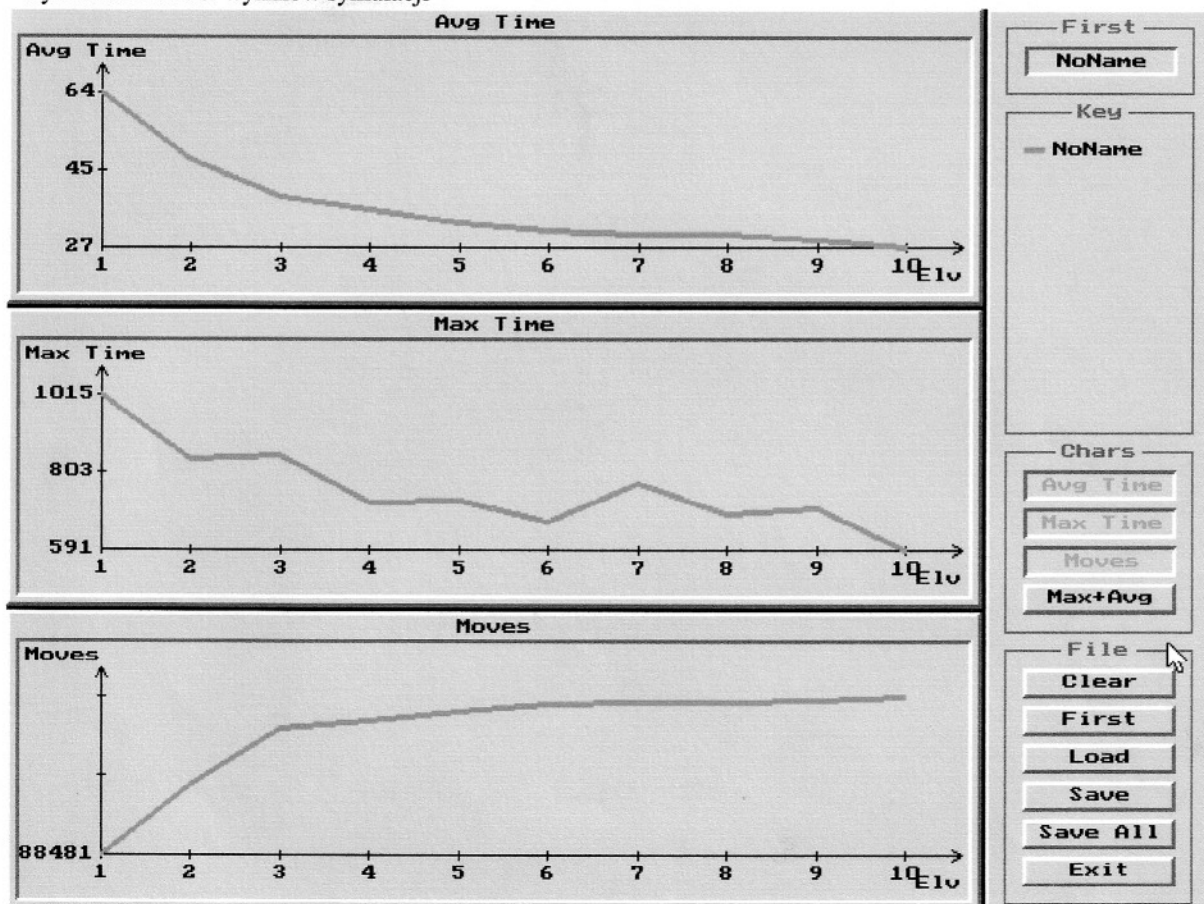
W czasie symulacji w tle paski postępu informują użytkownika o ilości pracy wykonanej przez program symulacyjny. Postęp paska **Elevators** obrazuje ilość zbadanych systemów, natomiast pasek **Timer** odpowiada czasowi pojedynczej symulacji. W każdym momencie działania symulacji w tle można ją przerwać klawiszem **Cancel**.

### 3.10. Wyniki symulacji

Wynikiem symulacji pełnej jest zestaw wykresów w zależności od ilości wind:

- **Avg Time** – średni czas oczekiwania na windę
- **Max Time** – maksymalny czas oczekiwania na windę
- **Moves** – liczba ruchów wind pomiędzy sąsiednimi piętrami

Rys. 3.10.1. Panel wyników symulacji



Menu wyników symulacji zawiera następujące informacje i funkcje:

- **First** – Okno informacyjne zawierające nazwę pierwszego zestawu wykresów.
- **Key** – Legenda. Zawiera nazwy wszystkich wczytanych wyników i przyporządkowane im kolory wykresów. Najeżdżenie myszką na pole zawierające nazwy powoduje wyświetlenie **okna parametrów** i zaznaczenie jednej z nazw. Kliknięcie na zaznaczoną nazwę powoduje wyłączenie wyświetlania odpowiadających jej wykresów i wyświetlenie zamiast nazwy napisu **-off-**. Kliknięcie na napis **-off-** powoduje włączenie wyświetlania wykresów. Jeśli myszka wskazuje na okno wykresu to w polu **Key** zamiast nazw wyników wyświetlane są wartości wyznaczonych funkcji dla ilości wind wskazanych przez myszkę.
- **Avg Time** – Klawisz włączający/wyłączający wyświetlanie wykresów średniego czasu oczekiwania.
- **Max Time** – Klawisz włączający/wyłączający wyświetlanie wykresów maksymalnego czasu oczekiwania.

**Moves** – Klawisz włączający/wyłączający wyświetlanie wykresów ilości ruchów wind.

**Max+Avg** – Klawisz włączający/wyłączający tryb wyświetlania wykresów czasu średniego i maksymalnego w jednym oknie.

**Clear** – Wymazanie z pamięci wszystkich wyników symulacji.

**First** – Wymazanie z pamięci wszystkich wyników symulacji za wyjątkiem wyników o nazwie podanej w oknie informacyjnym **First**.

**Load** – Odczyt z pliku wyników symulacji i dołączenie ich do wyników w pamięci.

**Save** – Zapis do pliku wyników symulacji o nazwie podanej w oknie informacyjnym **First**.

**Save All** – Zapis do pliku wszystkich aktywnych (wyświetlanych) wyników symulacji.

**Exit** – Wyjście z panelu wyświetlania wyników i przejście do panelu symulacji.

Rys. 3.10.2. Okno parametrów.

Fl	Mt	St	Me	Mf	Alg
10	5	5	10	50	LastFlr
10	5	5	10	200	LastFlr
10	5	5	10	200	Down-Up
10	5	5	10	50	NextFlr

Okno parametrów zawiera następujące informacje:

**Fl** (Floors) – Liczba pięter w badanym budynku.

**Mt** (Move Time) – Średni czas przejazdu windy między dwoma sąsiednimi piętrami.

**St** (Stop Time) – Średni czas postoju windy na piętrze.

**Me** (Men Elv) – Maksymalna liczba pasażerów w windzie.

**Mf** (Men Flr) – Maksymalna liczba oczekujących na piętrze.

**Alg** (Algorytm) – Nazwa algorytmu obsługi wind.

## 4. Moduł algorytmów

Moduł algorytmów **algs.pas** znajduje się w folderze **ALG**. Jest to biblioteka napisana w języku Turbo Pascal, kompilowana za pomocą standardowego kompilatora **tpc.exe** a jej wklejenie do programu symulacyjnego realizowane jest przez dołączony kompilator **elvcmp.exe**. Moduł algorytmów zawiera implementację algorytmów dostępnych w programie symulacyjnym. Może on być modyfikowany przez użytkownika. Umożliwia to edycję, usuwanie oraz dodawanie nowych algorytmów, dzięki czemu program symulacyjny jest bardziej elastyczny i uniwersalny. Implementowanie algorytmów wymaga znajomości języka Turbo Pascal. Błędy programisty na poziomie kompilacji powodują zaniechanie wklejenia zmodyfikowanego modułu, o czym powiadamia użytkownika kompilator. Błędy na poziomie programowym mogą spowodować nieoczekiwane i niepożądane skutki, dlatego zaleca się nie modyfikowanie nieopisanych fragmentów kodu modułu.

### 4.1. Składnia modułu algorytmów

```
{ $A+, B-, D+, E+, F-, G-, I+, L+, N-, O-, P-, Q-, R-, S+, T-, V+, X+, Y+ }
{ $M 50000, 0, 655360 }

unit algs ;

interface

uses
    vars ;

const

    algs_nr = liczba_algorytmów ;
    algs_names : array [1..algs_nr] of String =
        ('nazwa_algorytmu1', 'nazwa_algorytmu2', ...) ;

procedure algorytms ;
procedure reset_algs ;

implementation

{-----}
{ALGORYTMS IMPLEMENTATION}
{-----}

definicja_zmiennych_globalnych_modułu ;

procedure algorytm1 ( argumenty : id_argumentów ) ;
definicja_zmiennych_i_stałych ;
begin
    kod_procedury ;
end ;

{-----}
```

```

kody_innych_prcedur ;

{-----}
{ End of ALGORYTMS IMPLEMENTATION }
{-----}

procedure algorytms ;
begin
    case nr_alg of

{-----ALGORYTMS NUMBERS-----}

        1 : algorytm1 ;
        2 : algorytm2 ;
        pozostałe_procedury

{-----End of ALGORYTMS NUMBERS-----}

    end ;
end ;

{-----}
{  STARTUP }
{-----}

procedure reset_algs ;
definicja_zmiennych_i_stałych ;
begin
    kod_procedury_reset_algs ;
end ;

{-----}
begin end.
{-----}

```

## 4.2. Edycja modułu algorytmów

### 4.1.2. Liczba algorytmów

Przy usuwaniu lub dodawaniu nowych algorytmów konieczna jest zmiana wartości stałej `algs_nr`. Odpowiada ona za ilość algorytmów do wyboru w programie symulacyjnym. Przykładowo dla dwóch algorytmów linia ta będzie miała postać:

```
algs_nr = liczba_algorytmów ;
```



#### 4.2.2. Nazwy algorytmów

Tablica stałych `algs_names` zawiera nazwy algorytmów, stanowiące identyfikatory algorytmów w programie symulacyjnym. Nazwy należy wpisywać w apostrofach i oddzielać przecinkami. Liczba wprowadzonych nazw musi być równa wartości stałej `algs_nr`. W przeciwnym razie wystąpi błąd na poziomie kompilacji. Nazwy należy wprowadzać w kolejności ustalonej po instrukcji `case nr_alg of` w procedurze `algoritms`. Przykładowo dla dwóch algorytmów o nazwach kolejno **First** i **Second** definicja tablicy będzie wyglądała następująco:

```
algs_names : array [1..algs_nr] of String =  
    ('First', 'Second') ;
```

#### 4.2.3. Implementacja algorytmów

Implementacja algorytmów następuje po słowie kluczowym `implementation` w postaci procedur o dowolnych nazwach i parametrach, przy czym stosowanie parametrów nie jest konieczne. Dozwolone jest definiowanie własnych zmiennych globalnych modułu algorytmów, których wartości początkowe przypisać można poprzez procedurę `reset_algs`.

#### 4.2.4. Włączanie i wyłączanie algorytmów

Włączanie algorytmów następuje poprzez wpisanie bądź usunięcie nazw procedur opisujących algorytmy i przydzielenie im kolejnych numerów po instrukcji `case nr_alg of` w procedurze `algoritms`. Nazwy tu wpisane muszą być identyczne z nazwami wyżej zdefiniowanych procedur i muszą zawierać odpowiednią liczbę parametrów typu określonego w definicji. Przydzielane numery powinny być kolejne i powinny odpowiadać kolejności nazw w tablicy nazw `algs_names`. Ilość włączonych algorytmów powinna być równa wartości stałej `algs_nr`.

#### 4.2.5. Wartości początkowe zmiennych globalnych modułu algorytmów.

Wartości początkowe zmiennych zdefiniowanych przez użytkownika mogą być ustawiane w procedurze `reset_algs`. Jest ona wywoływana przy każdym wciśnięciu klawisza **Reset** w programie symulacyjnym.

#### 4.2.5. Sterowanie windami

Sterowanie windami realizowane jest przez ustawianie wartości wektora windy. Wektor windy opisany jest przez tablice:

```
elv_vec : array [1..N] of Integer ;  
elv_to   : array [1..N] of Integer ;
```

Tablica `elv_vec` opisuje kierunek poruszania się *i*-tej windy i może przyjmować następujące wartości:

- 0 – winda nie porusza się (jest pusta)
- 1 – winda porusza się w dół
- 1 – winda porusza się w górę

Wartości tej tablicy zmieniają się automatycznie:

- Gdy zatrzymująca się winda jest pusta lub wysiadają z niej wszyscy pasażerowie `elv_vec` zmienia się na 0.
- Gdy do stojącej windy wsiadają pasażerowie `elv_vec` zmienia się na wartość zgodną z kierunkiem ich jazdy

Wartości tablicy `elv_to` są równe numerom najbliższych pięter  $\langle 1, M \rangle$ , na których winda ma się zatrzymać w celu zabrania pasażera. W celu wysadzenia pasażera winda zatrzymuje się automatycznie. Wartość `elv_to` jest zerowana, gdy winda zatrzyma się na piętrze wskazanym przez tę zmienną. Ustawianie `elv_to` realizowane jest tylko przez procedurę obsługi algorytmu.

#### 4.2.6. Status windy i piętra

Procedura obsługi algorytmu powinna ustawiać wektory wind na podstawie danych o statusie windy i piętra. Informacje te zawarte są w zmiennych globalnych:

**N** : Integer – liczba wind, wartość z zakresu  $\langle 1, 20 \rangle$   
**M** : Integer – liczba pięter, wartość z zakresu  $\langle 1, 20 \rangle$

**stop\_time** : Integer – średni czas postoju windy na piętrze  
**move\_time** : Integer – średni czas przejazdu windy pomiędzy sąsiednimi piętrami  
**max\_men** : Integer – maksymalna liczba osób w windzie  
**max\_fmen** : Integer – maksymalna liczba osób na piętrze

**elv\_pos** : array [1..N] of Integer – numer piętra, na którym aktualnie znajduje się winda  
**elv\_men** : array [1..N] of Integer – liczba pasażerów w windzie  
**elv\_mento** : array [1..N, 1..elv\_men[i]] – status pasażerów,  
                  `elv_mento[i, j]` określa numer piętra, na które jedzie *j*-ty pasażer w *i*-tej windzie

**elv\_times** : array [1..N,1..elv\_men[i]] – status pasażerów,  
 elv\_times[i,j] określa czas oczekiwania na dojazd j-tego pasażera w i-tej windzie

**elv\_time** : array [1..N] – czas jazdy windy i-tej od ostatniego postoju, gdy  
 elv\_stat[i]=True lub czas postoju windy nie większy niż stop\_time, gdy  
 alv\_stat[i]=False

**elv\_stat** : array [1..N] of Boolean – status windy  
 - elv\_stat[i] = True – oznacza gotowość windy do jazdy  
 - elv\_stat[i] = False – czas postoju windy na piętrze jest krótszy od  
 stop\_time

**attr\_ar** : array [1..M] of Real – częstotliwość odwiedzania piętra,  
 prawdopodobieństwo wystąpienia zgłoszenia na i-te piętro wynosi  
 attr\_ar/sum\_attr, gdzie sum\_attr jest sumą attr\_ar[i] dla i=1..M

**attr\_lv** : array [1..M] of Real – średnia liczba zgłoszeń na minutę dla  
 piętra

**floors\_down** : array [1..M] – liczba osób oczekujących na piętrze na windę w  
 dół

**floors\_up** : array [1..M] – liczbe osób oczekujących na piętrze na windę w  
 górę

**floors\_downto** : array [1..M,floors\_down[i]] – status oczekujących,  
 floors\_downto[i,j] oznacza numer piętra, na które chce dojechać j-ty oczekujący  
 na i-tym piętrze na windę w dół

**floors\_upto** : array [1..M,floors\_up[i]] – status oczekujących,  
 floors\_upto[i,j] oznacza numer piętra, na które chce dojechać j-ty oczekujący na i-  
 tym piętrze na windę w górę

**floors\_downtimes** : array [1..M,floors\_down[i]] – status  
 oczekujących, floors\_downtimes[i,j] zawiera czas oczekiwania na windę w dół j-  
 tego oczekującego na i-tym piętrze

**floors\_uptimes** : array [1..M,floors\_up[i]] – status oczekujących,  
 floors\_uptimes[i,j] zawiera czas oczekiwania na windę w górę j-tego  
 oczekującego na i-tym piętrze

#### 4.2.7. Uwagi

W module algorytmów nie należy:

- zmieniać ani usuwać nazwy modułu unit algs
- usuwać z vars listy dołączonych modułów uses
- usuwać procedur algorytms i reset\_algs
- parametryzować procedury algorytms i reset\_algs
- usuwać begin end. na końcu modułu

Każda z powyższych czynności zakończy się błędem kompilacji.

W procedurach obsługi algorytmów oraz w reset\_algs nie jest wskazana zmiana wartości zmiennych globalnych za wyjątkiem elv\_vec i elv\_to, które są do tego celu przeznaczone.

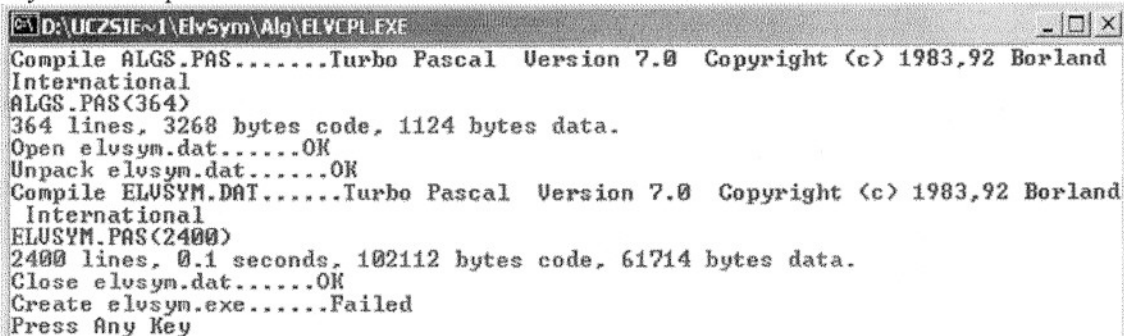
## 5. Kompilator

Kompilator **elvcmp.exe** znajduje się w folderze **ALG**. Jego zadaniem jest wklejenie zmodernizowanego modułu algorytmów do programu symulacyjnego. Po jego uruchomieniu wykonywane są następujące czynności:

- kompilacja modułu algorytmów **algs.pas**
- otwarcie pliku danych **elvsym.dat**
- dekompresja pliku danych
- wklejenie modułu algorytmów do programu symulacyjnego i kompilacja programu
- zamknięcie pliku danych
- utworzenie **elvsym.exe**

Informacje o poprawności wykonania powyższych czynności wyświetlane są w oknie MS-DOS. Jeśli program **elvsym.exe** w katalogu głównym pakietu już istnieje zostaje wyświetlona informacja „Create elvsym.exe...Failed” a program tworzony jest w folderze **ALG**. Kompilator kończy działanie po naciśnięciu dowolnego klawisza.

Rys. 5.1. Kompilator



```
D:\UCZSIE~1\ElvSym\Alg\ELVCPL.EXE
Compile ALGS.PAS.....Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland
International
ALGS.PAS(364)
364 lines, 3268 bytes code, 1124 bytes data.
Open elvsym.dat.....OK
Unpack elvsym.dat.....OK
Compile ELUSYM.DAT.....Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland
International
ELUSYM.PAS(2400)
2400 lines, 102112 bytes code, 61714 bytes data.
Close elvsym.dat.....OK
Create elvsym.exe.....Failed
Press Any Key
```