Лабораторна робота №6 ОСОБЛИВОСТІ РОБОТИ ФУНКЦІЇ PRINTF

**Мета:** Дослідити особливості роботи функції printf.

**Індивідуальне завдання:** створити програму з використанням функції printf, що дозволяє:

* зчитати дані змінних стеку поточної функції (що були декларовані до моменту виклика функції)
* зчитати дані заданої ділянки пам’яті, до якої має доступ поточна

програма.

ХІД РОБОТИ

Виконання лабораторної роботи проводилося у сфері розробки Eclipse IDE. Далі наведено код програми, що використовувався.

/\*\*

* nain.cpp

\*

* Created on: 28 февр. 2019 г.
* Author: pavlova-mv

\*/

#include <stdio.h> #include <cstdlib> #include <cstring>

void printStack(int stack[], int stackSize) { for (int i = 0; i < stackSize; i++) {

printf("stack %d: %d\n", i, stack[i]);

}

}

void printCell(char\* someMemoryCell) { printf("memory cell: %s\n", someMemoryCell);

}

int main() {

setvbuf(stdout, NULL, \_IONBF, 0); printf("=================Part1\n"); int size = 10;

int \*newStack = new int[size];

for (int i = 0; i < size; i++) {

newStack[i] = 0 + rand() % 100;

}

printStack(newStack, size);

printf("=================Part2\n"); char\* cell = new char[15]; printCell(cell);

cell = "some text"; printCell(cell);

delete[] newStack; delete cell; system("pause"); return 0;

}

У першій частині програми продемонстровано виведення даних зі стеку, що було створено до виклику функції з використанням printf().

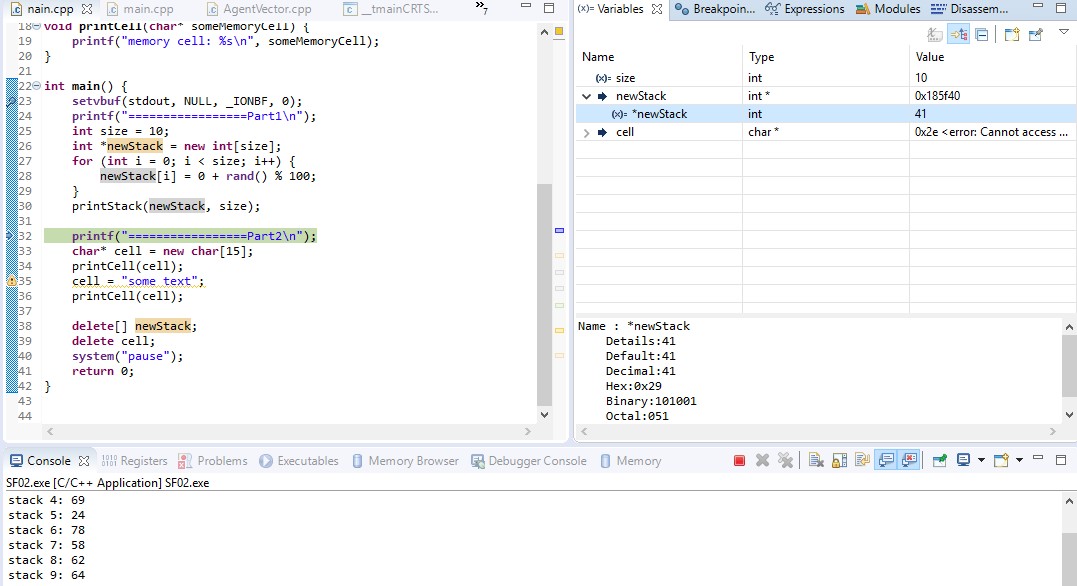


Рисунок 1 — Перша частина програми (створення та виведення стеку)

Червоним кольором на рис.1 показано типи даних, що використовувались у стеку, а також адреси пам’яті.

У наступній частині програми проводиться демонстрація виведення деякої ділянки пам’яті. Для найбільш чіткого відображення назначення функції printf() спочатку ми намагаємося вивести зміст ділянки пам’яті бзе поперднього запису у неї, а потім після запису. Адреси ділянки пам’яті можна побачити на рис.2, а результат зчитування на рис.3

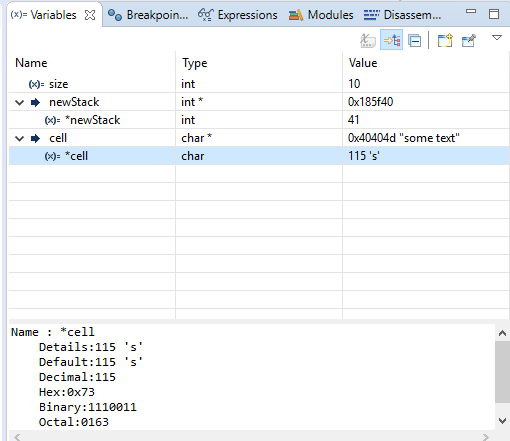


Рисунок 2 — Адреси

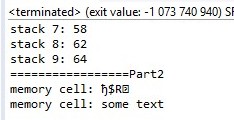


Рисунок 3 — Результат другої частини програми

Однією з особливостей використання функції printf() y Eclipse є те, що необхідно перенаправляти буферізацію потоків введеня/виведення. Тож для отримання результатів у консолі необхідно використовувати setvbuf(stdout, NULL, \_IONBF, 0) або fflush(stdout).

ВИСНОВКИ

В результаті виконання лабораторної роботи було досліджено особливості роботи функції printf.