YAUHENIYA PADBIAROZSKAYA

308295

Lab 4

1. Implementacja własnej powłoki
    - funkcje systemowe:
      waitpid(2) – zawiesza wykonywanie bieżącego procesu dopóki potomek określony przez pid nie zmieni swojego stanu (zakończy działanie,zostanie zatrzymany przez sygnał).
      fork(2) – tworzy process potomny posiadający własne PID i PPID.
    - funkcje operujące na stringach:
      strcmp(3) – porównuje 2 stringi, zwraca 0 jeśli są równe, <0 jeśli s1<s2, >0 jeśli jest na odwrót
      strtok(3) – rozbija string na kilka tokenów na podstawie podanego jako 2. argument łącznika.
      strchr(3) – zwraca wskaźnik do pierwszego wystąpienia char w podanym stringu.
    - funkcje zarządzające pamięcią:
      realloc(3) – zmienia rozmiar bloku pamięci wskazywanego przez ptr na size bajtów.
      free(3) – zwalnia obszar pamięci wskazywany przez pointer.
    - inne funkcje biblioteczne:
      exec(3) – zastępuje w pamięci obraz aktualnego procesu obrazem nowego procesu.
      execvp(3) – udostępnia tablicę wskaźników do listy argumentów programu. Pierwszy argument wskazuje na nazwę pliku. Tablica wskaźników musi być zakończona wskaźnikiem NULL.

   2) początkwo znak zachęty nie czekał na zakończenie bieżacego procesu, żeby to naprawić trzeba było zmodyfikować funkcję *executecmds().* Dla tego celu wykorzystujemy funkcję *waitpid(),* która wstrzymuje proces macierzysty, który musi czekać dopóki proces potomny nie zmieni swojego stanu. Gdy dziecko pomyślnie skończy działanie, ustawiamy *proces=0* i wypisujemy kod wyjścia procesu.
   3) żeby polecenie *exit* działało poprawnie, za pomocą komendy *strcmp()* sprawdzamy czy wpisana jest właśnie ta komenda, potem dla procesu głównego wywołujemy *exit()*, co sprawia, że wychodzimy z pętli w *main*.
2. Ustawiania limitów procesów
    - funkcje systemowe:
      getrlimir(2) -- pobiera limity zasobów zdefiniowane w strukturze rlimit (miękkie i sztywne ograniczenia)
      setrlimit(2) – ustawia limit zasobów.
      _exit(2) – zakończenie procesu, bez wywołania kodu czyszczącego (w odróżnieniu od *exit*)
    - *argc i argv* – liczba argumentów przekazanych do programu i odpowiednio wektor argumentów, gdzie argv[0] – nazwa programu

- *atoi(3)* -- jako argument pobiera liczbę w postaci *const char**, a następnie zwraca jej wartość w formacie int. Funkcja kończy wczytywać znaki w momencie napotkania znaku, który nie jest cyfrą.
- *struct rlimit* – zdefiniowane ograniczenia miękkie i sztywne. Ograniczenie miękkie jest wartością zasobu wymuszoną przez jądro. Ograniczenie sztywne -- wartość maksymalna dla ograniczenia miękkiego: proces nieuprzywilejowany może sobie ustawić ograniczenie miękkie w zakresie od 0 do ograniczenia sztywnego oraz (nieodwracalnie) obniżyć swoje ograniczenie sztywne. Proces uprzywilejowany może dowolnie zmieniać każdą z wartości ograniczenia.